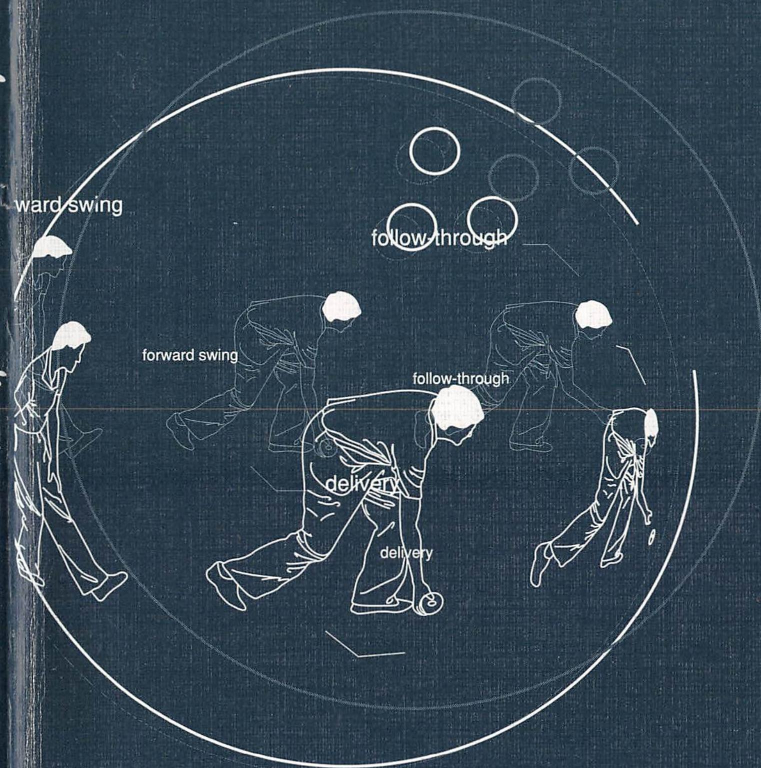
通産省情報処理技術者試験
1998年8月1日発行／第14巻第8号
毎月1回1日発行

情報システム辞典

一九九八年度版



- システム開発とプログラミング
- ファイルとデータベース
- アルゴリズムとデータ構造
- 産業社会と情報化
- 情報処理システム

学研 合格情報処理　8月号第2付録

合格情報処理 — 8月号第2付録 — 一九九八年八月一日発行／第十四巻第八号 毎月一回一日発行

情報システム辞典 一九九八年度版

学研



情報システム辞典

Gakken

PRINTED IN JAPAN

通産省情報処理技術者試験対策

情報システム辞典 1998年版

情報処理技術を学ぶうえでも，
語るうえでも
技術用語を正しく
理解しておくことは大切なことです。
情報処理用語は
簡単な語でも日常語とは異なり，
特定の意味で用いられたり，
特殊な使われ方をすることがあります。
これらを正しく知ることは，
勉強のためにも，将来の仕事のためにも
必要なことです。

ア
カ
サ
タ
ナ
ハ
マ
ヤ
ラ
ワ
英数字

学研

◆採択用語について◆

★本辞典は，第二種共通カリキュラム・シスアド共通カリキュラムに示された重要用語を中心にその周辺の用語まで収録したものです。

★「合格情報処理」別号付録の「情報処理基本用語辞典」（ハード／ソフト／情報処理一般），「パソコン・シスアド用語辞典」，「ネットワーク用語辞典」とともにご利用いただければ幸いです。重複を極力抑えましたので，ご併用いただければ幸いです。

◆用語の表記と掲載順◆

★見出し語の表記，訳語は一定していない現状ですが，カリキュラムの表記，または共通テキストの表記に準じました。なお不十分と思われる語は，慣用の語や別の見出し語として収録しました。

★用語は五十音順に配列しました。片仮名の長音は前の字の母音とみなします（つまり長音符号を無視しない）。国語辞典の標準的な配列法に準じました。
また，拗音，促音，濁音，半濁音は清音と見なします。
（例：「モジュール」は「もしゆうる」，「バーチャル」は「はあちやる」の読み）

★英語・英略語は混在させ，アルファベット順に配列しています。大文字と小文字は区別しません。

★見出し語に続く矢印は，その項目に解説があることを示します。解説末尾の矢印は，その項目をあわせて参照すると理解が深められることを示します。

（同）＝同義語，言い換え語	（類）＝類義語，ほぼ同義語
（対）＝反対語，対照語，対語	（参）＝参照項目，参考項目
（関）＝関連語，同族語	（例）＝例示項目

★見出し語中の／や：は，言い換え，または略語であることを示します。

◆執　筆◆（五十音順）

鮎沢　正治	新井　忠明	飯島　秀之	池田　美保	板野　敏貴
稲葉　久男	内田　義也	河内　栄一	川崎　清隆	川真田政夫
木幡　　悟	坂井　浩二	酒井　力郎	澤村　英仁	進藤　良雄
鈴木真由美	高澤　洋子	高橋　　勇	高橋　健吉	滝口　郁志
鶴岡　玄光	寺沢　康夫	豊住　昌雄	西村　修一	長谷川弘子
秦　　　浩	林　　篤	増井　和也	真々田　稔	守　　章彦
山口　行広	山本　和明	脇山　久		

■表紙デザイン／石井知哉　■編集協力・図版／(有)メビウス

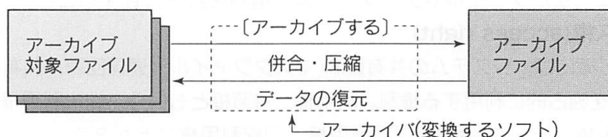
ア

アーカイブ(archives)

保管のために複数のファイルをひとつにまとめたり，データ圧縮を行うこと。アーカイブしたデータは，必要に応じて再現(復元)する。これらの操作を行うソフトウェアをアーカイバ(archiver)という。

アーカイブファイル(archive file)

原典のアーカイブ対象ファイルからアーカイブによって作り出し，原典を再現するのに十分なデータを含むファイル。後の利用のために組織的に作成し，維持し，一定の期間，保存(保管)する。記憶階層の最下位層に相当する。例えば，磁気テープで保存したファイルは，必要に応じて磁気ディスクに移して使用する。



アーキテクチャ(architecture)

本来は建築学の分野で，建築様式，建築物の意味をもつ語。コンピュータの分野では，ハードウェアやネットワークの構成様式の意味に用いる。

狭義には，中央処理装置について，データや命令の形式，アドレス指定，割込み方式など，プログラマが必要とするシステムの特性，特徴の集まりのことをいう。

アーギュメント(argument)

→(同)引数

あいまい検索(ambiguous searching)

データベースや文書中の文字列データの検索を行うとき，文字コードの一致を前提としないで許容範囲を広げて文字列を検索すること。例えば，英字の大文字と小文字，全角文字と半角文字，ひらがなとカタカナの区別をしない検索をさす。さらに許容範囲を広げた類義語検索，具体的な文字列を指定しないワイルドカード検索，文字の位置などが指定できる正規表現検索などがある。

アウトソーシング(out sourcing)

広義には，事業の効率化や制度改革のために，自社外の経営資源，人的資源を活用すること。情報処理の分野では，企業内の情報システムの開

発，運営，管理業務を，コンピュータメーカやソフトウェア会社など外部の企業に一括して委託すること。企業の情報システムに関する作業を一括して代行するサービスを利用することも含む。

その対象は新システムの企画・開発，ハードウェアやソフトウェアの選定・調達，システム構築，運用管理，保守，バージョンアップ作業，さらにネットワークの監視，支援サービス，要員の育成・確保など広範囲におよぶ。したがって，アウトソーシングを請け負う企業は，多方面の技術力や信用が必要である。日本では1994年，通産省が「特定システムオペレーション企業認定制度」を設けて，事業の円滑化を図っている。

アクセス(access)

①CPUが，プログラムの実行中に主記憶装置や外部記憶装置などのデータを読み出したり，データを書き込んだりすること。

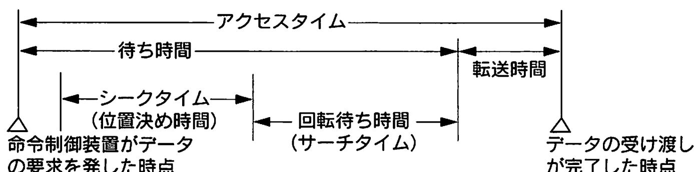
②ユーザがファイルやデータベースの情報を読み書きすること。

アクセス権(access right)

広義には，システムの共有資源(マスタファイル，通信回線，共有メモリ等)を独占的に利用する権利。利用権，参照権ともいう。端末利用権，CPU利用権，データ利用権，OS利用権，回線利用権などがある。

アクセス時間(access time)

制御装置がデータを補助記憶装置の指定する記憶場所に書き込んだり，指定する記憶場所から読み取る要求を出してから，データの転送が完了するまでの時間。呼出し時間ともいう。コンピュータの処理速度を表す基準のひとつである。



アクセス制御(access control)

コンピュータシステムや資源に対する利用要求を管理し制御すること。具体的には，アクセスのログを残し，利用者のアクセス要求とアクセス権を検証し，不当なアクセスを制限し，警告を発するなどの処理を行う。アクセス制御は誤操作によるデータ破壊やシステムの不正利用を防ぐためだけでなく，障害の復旧，ユーザへの情報提供に重要な役割を担っている。

アクセス方式(access method)

プログラムによるデータやレコードの読込み，書出し，およびデータやレコードの取扱い方法。アクセス法ともいう。レコードの集合であるファイルは，記憶媒体に記録されるとき各種の編成方法(論理構造)が与えられるが，それぞれに応じて適切なアクセスを行うアクセス法がオペレーティングシステムにより提供される。順次アクセス(SAM)，ランダムアクセス(DAM)，索引順次アクセス(ISAM)，仮想記憶における索引順次アクセス(VSAM)などに分けることができる。

アクセスモード(file access mode)

Cの用語で，ファイルオープン(fopen)時に指定するファイルのアクセスモードのこと。ファイルの操作(読み／書き，テキスト／バイナリ)に応じて指定する。

アスキーファイル(ASCII file)

テキスト(英数字)，若干の制御コードなど1バイトのASCIIコードだけで構成されるファイル。→(類)テキストファイル，(参)アスキーセーブ

アスキー符号(ASCII code)

→(同)ASCIIコード

アセンブラ(assembler)

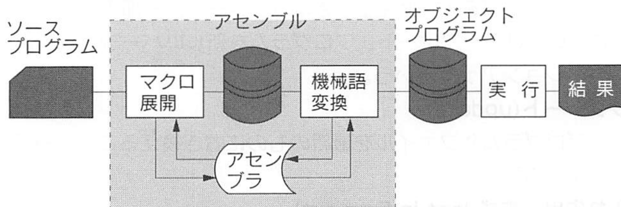
アセンブラ言語で書かれたソースプログラムを，機械語のオブジェクトプログラムに翻訳する言語処理プログラム。アセンブラの命令語には，

(1)実行時の命令コードと1対1に対応する命令語

(2)アセンブル過程に指示を与えるアセンブリ命令

(3)アセンブル過程で一連の機械語で置き換えるマクロ命令

などがある。→(参)マクロアセンブラ



アセンブラ言語(assembly language)

2進数による機械語の命令を書くわずらわしさを解消するために作られた言語。例えば，加算はA，ロードはLというように記号(表意コード)を用いて書き表す。記号言語ともいう。高水準言語と異なり，コンピュータの



機種に依存した記述になる。

アセンブル(assemble)

アセンブラ言語で書かれたソースプログラムをアセンブラによりオブジェクトプログラム(機械語コード)に翻訳する作業。→(関)翻訳

値(value)

順序のある1組の数字(数値), 順序のある1組の論理値(論理値), 順序のある1組の文字(文字列)など, 表計算ソフトのセルやプログラムの変数に代入できるデータ。値は, 整数, 実数, 文字などの型をもち, 長さに制約があり, 指定された内部表現により, セル名や変数名で識別できる記憶場所に格納する。型や長さが未指定のワークシートを使用する際は指定する必要がある。

値呼出し(call by value)

手続きや関数を呼び出して使用するとき, パラメタの引き渡しを仮引数と実引数の対応で行うこと。Cでは値渡しという。

値渡し(call by value)

Cにおける基本的な関数呼出し方法で, 引数はスタックを経由して値が関数に渡される。この場合, 値を渡された関数の中で値を書き換えても, 呼び出した側の値に影響はない。→関数呼出し

アップグレード(upgrade)

ハードウェアやソフトウェアを高機能な製品に交換したり, 拡張したりすること。強力なマイクロプロセッサや周辺機器を加えたり, バージョンアップしたプログラムを入れたりして, システムを高機能化, 強化すること。比較的よく行われるのがOSのアップグレードである。Windows3.1からWindows95への移行などはアップグレードの典型である。特に, 全部のハードウェアを入れ換えるときはリプレース, 一部のファイルの置き換えについてはアップデート, プログラムを新しいバージョンに代えることはバージョンアップという。

アップデート(update)

プログラムやファイルを最新のものに置き換えること。→(同)更新, ファイルの更新

後入れ先出し方式(last in first out)

→(同)LIFO

後書きラベル(trailer label／end of file label：EOF)

磁気テープにデータを記録する際に, ファイルの最後に書き込む特別なレコード。ファイルに記録されたレコード数やブロック数などを記録し,

そのファイルを入力ファイルとして使うときに，すべてのレコードを確実に読み込んだかどうかなどの検査に使う。

①ファイル終わりラベル。ファイルの末尾に置く内部ラベル。ファイルの終わりを識別する。

②磁気テープでは終端マーカをさす。

アトミックアクション(atomic action)

コミットメント制御において制御対象となる処理単位。データベースの更新のように，処理途中で失敗した場合は処理以前の状態に戻さなければならない性質をもつ完結したひとつの処理をいう。トランザクション処理では，ひとつのトランザクション単位がひとつのアトミックアクションに相当する。なお，トランザクション処理以外にも同様の性質を要求する処理があるため，汎用性のあるアトミックアクションという言葉を使っている。原子動作ともいう。

アトリビュート(attribute)

コンピュータで扱うデータやファイルがもつ性質のこと。属性ともいう。例えば，次のような性質を属性という。

(1)ディスプレイに表示するデータは，対応する属性によって高輝度，反転・明滅などをする。字体や大きさ，表示色なども属性である。画像処理では，図形の解像度，座標の表し方，色などの情報。

(2)レコードの場合には，レコード長，レコード形式，データセット名，ボリューム識別番号，用途，作成日などを属性として記録する。

→(類)ディレクトリ

(3)ファイルアクセスの場合には，読み出し専用，システム専用，秘匿などの情報が属性となる。

(4)データベースの関係モデルを表現した二次元表の列のこと。このとき行はタプル(tuple)という。→(参)関係データベース，組

アドレス(address)

主記憶装置，磁気ディスク記憶装置などの中の記憶場所を示すための識別文字の集まり。番地ともいう。記憶装置を多数の電子的な記憶単位の集合として扱い，各単位を識別するために一連の番号(絶対アドレス)を付けておき，読み取ったり書き込んだりする際にはこの番地を使って記憶装置内のプログラムやデータを指定する。通常 1ワード分が 1番地分になる。ソースプログラム内では絶対アドレスのほか，ラベル名，記号アドレスで表現することもある。→(関)絶対アドレス，相対アドレス

アドレス空間(address space)

中央処理装置(CPU)がアクセス可能な記憶装置の領域。その物理的な大きさは主記憶装置のアドレス情報を制御するビット数で決まる。仮想記憶方式ではプログラム実行時に論理的なアドレス空間を主記憶装置のアドレス空間に変換することによって，実アドレス空間より大きい仮想アドレス空間を使用できる。→(参)仮想記憶

アドレス指定(addressing)

コンピュータの機械語レベルで，命令をどのデータに作用(オペレート)させるか決定すること。機械語の命令は，コンピュータにデータ処理の種類を指定する命令部と処理対象のデータの所在を指定するアドレス部とからなっている。アドレス指定の方式には，(1)直接アドレス指定，(2)間接アドレス指定，(3)即値アドレス指定などがある。

穴埋め

レコードを固定長にするために，短いデータの末尾などに空白文字などを付加すること。

アナライザ(analyzer)

分析器。解析器。微分解析器，ディジタル微分解析器，回路網解析器など多種多様の機器がある。

アプリケーションエンジニア(application engineer)

個別アプリケーションのシステム開発にかかわり，対象業務の分析，要求定義を行い，外部設計書の作成を行う技術者。情報システムの構築に際し，対象業務(アプリケーション)を分析，モデル化し，情報システムを設計，構築する。

アプリケーションプログラム(application program)

現実的な業務で使用するデータを処理するプログラム。オペレーティングシステム，ミドルウェア(サービスプログラムやプログラム言語など)，共通ソフトウェア(日本語FEPや通信ソフトウェアなど)に対比する用語。

具体的には，在庫管理，工程管理，販売予測など特定の目的でデータ処理するプログラムをさす。応用プログラム，あるいは適用業務プログラムともいう。パソコンの場合，コンピュータメーカやソフトウェア会社が作成し，使用説明書とともに販売している。このように既製品化したプログラムのことを，特にアプリケーションパッケージ，または略してアプリケーションという。ユーザの環境や要求に合わせるためにカスタマイズという作業が必要になる場合がある。

→(対)システムプログラム

あふれ域(overflow area)

→(同)オーバフロー領域

アベイラビリティ(availability)

→(同)可用性

アベンド(abend／abnormal end of task)

タスクの異常終了。タスクがエラーを起こして処理を続けられなくなり終了すること。または，他のタスクやジョブ等に悪影響を与えるため，制御プログラムによって異常終了させられること。

アボート(abort／abnormal termination)

コンピュータで実行中の処理が，ハードウェアの異常やプログラムのエラーにより，続行が不可能，または望ましくない状態になったとき，OSまたはオペレータにより処理を中断すること。異常終了，停止ともいう。プログラムが異常終了すると，一般的なプログラムの後処理(オープン中のファイルのクローズ，出力中のデータのフラッシュ等)は行われない。

アメダス(Automated Meteorological Data Acquisition System：AMeDAS)

地域気象観測システム。気象庁が1974年から全国約1300か所に設置した自動気象観測地点から降水量，風向風速，気温，日照時間などをNTTの通信回線で集信するシステム。

アラーム(alarm)

コンピュータシステムで異常状態が発生したことを知らせる警報。ランプの点滅，ビープ音の発生，画面へのメッセージ表示などによってユーザの注意，対応を喚起する。

アラインメント(境界調整)(alignment)

データオブジェクトの型によって，特定のアドレスに記憶領域の境界の調整が行われること。通常，アドレスがある整数の倍数になるように調整される。この境界の調整の仕方(アドレッシング方式)は，実行環境に依存する。アラインメントを行うと，メモリの使用効率は悪化するが，アクセス速度は向上する。

アルゴリズム(algorithm)

与えられた問題を機械的に解くために組み立てられた，あいまいさのない解法。算法ともいう。アルゴリズムは，順序，分岐，繰返しをもつ有限個の操作(命令の組合せ)で定義してあり，文や流れ図でも表現できる。計算機で解くためにプログラム言語で表現するとプログラムになる。プログラムは，有限回の適用(必ず実行が終了する)によって所定の解答を得る。



広義には処理手順(手続き)一般をさす。→(類)手続き，段階的詳細化

アルファニューメリック(alphanumeric)

英数字のこと。英字および数字を含み，場合によっては制御文字，特殊文字，間隔文字(空白文字)を含む1バイト(半角)の文字集合。事実上，ソースプログラムを記述するときに使用できる文字である。

アルファニューメリックチェック(alphanumeric check)

入力データのチェックの一手法。データは英数字と規定した項目が英数字であるか調べること。例えば，英数字だけから成る商品コードに誤って英数字以外の文字が混入するのを防ぐために行う。

アルファ版(alpha version／α version)

開発中のソフトウェアで，ある程度完成に近い段階で，テストのために社内のユースに提供するバージョン。さまざまな機種や局面で使ってもらい，意見や感想，バグ情報を集めたりする。→(参)ベータ版

アレイコンピュータシステム(array computer system)

データの高速処理の目的で，アレイプロセッサを中核に構成したコンピュータシステム。→(参)アレイプロセッサ

アレイプロセッサ(array processor：AP)

一般にベクトル形のデータの各要素ひとつひとつに同じ操作を施す作業を並列的に実行できるように，多数の演算装置を配列状に配置したプロセッサのことをいう。並列処理の方式により，空間的並列方式と時間的並列処理方式のふたつに大別できる。→(参)パイプライン処理

アローダイアグラム(arrow diagram)

日程管理手法のひとつであるPERTで使われるもので，日程を計画するために使用される図。矢線図ともいう。作業を進める順に，作業を矢線(アクティビティ)で記入し，作業と作業を丸印(イベント)で結ぶことによって，それぞれの作業の開始または終了を表す。作業の相互関係とネックとなる作業を確認することができる。→(類)PERT，グラフ

アロケーション(allocation)

プログラムを実行する際，ファイルや記憶領域を必要に応じて割り当てる。このようなファイルや記憶領域を割り当てることをアロケーションという。マルチプログラミングでは，いつ，どの番地にプログラムを格納するか，その場にならないと決まらないので，プログラムは相対番地で作成する。OSは実行時に主記憶上の空き領域内にプログラムを格納する。

アンインストール(uninstall)

使用中のハードディスク装置から不要なソフトウェアを削除するとともも

に，初期化ファイルやショートカット，メニューなどの内容をインストール前の状態に戻すこと。アンインストーラというソフトにより処理する。最近のソフトウェアは多数のファイルから構成されており，使用のつどシステム用のファイルが書き換えられていることもある。手作業でインストール前の状態に戻すのは容易ではなく，この作業の自動化機能を有するアンインストーラを使用して行うのが通例である。

アンケート調査(questionnaire survey)

システム分析の際，システム化に対する意見(意向)調査のために行う文書と記述による調査。調査対象の広さで，悉皆(※しっかい)調査(全員調査)，標本調査(選んだ一部の人の調査)がある。また，解答形式では，多肢選択式，自由回答式などがある。標本抽出なら回答者に偏りがないように，多肢選択式では先入観が入らないように，誘導的な設問をしないようにする。また，回答者の属性(職種・職位・年齢・性別など)が把握できることも大切である。

安定な整列アルゴリズム(stable sort argorithm)

整列アルゴリズムの呼称で，同じキーをもつデータが二つ以上あったとき，それらのデータは，整列後ももとの配列にあった順序の通りに並べるアルゴリズム。例えば，氏名の五十音順に並んだ名簿ファイルを生年月日をキーに並びかえたとき，生年月日が同じ人が6人いたとすれば，その6人の並びの部分はやはり氏名の五十音順に整列させるアルゴリズム。具体的にはビンソート(bin sort)というアルゴリズムがある。

アンデリート(undelete)

誤って削除したファイルを復元すること。復元には削除後，そのファイルを記録した領域に別のファイルがオーバライトされていないことが必須要件になる。一般には，ファイラ(ファイル操作ユーティリティ)の機能を利用する。Windows95では，ファイルの削除はアイコンのごみ箱への投入で指示するが，ごみ箱を｢空にする｣操作を行うまでは，ごみ箱ディレクトリ(ごみ箱フォルダ)にファイルが残っているため，再生が可能である。

アンドゥ(undo)

直前に行った処理を取消し，もとの状態に戻す指示。ワープロなどの対話型アプリケーションソフトには，誤操作によるデータの破壊や消滅を防ぐため，この機能が具備されている場合が多い。→(参)バックアップ

アンバンドリング(unbundling)

ハードウェアとソフトウェアの価格を分離して販売すること。1960年代までのコンピュータの普及期の販売では，ソフトウェア，教育，保守サ

ービスなどはハードウェアの付属サービスと見なされ，ハードウェア価格に組み込まれていた(バンドリング)。ソフトウェア価格(価値)の比重の高まりにともない，ソフトウェアも独立して販売されるようになった。現在のソフトウェアが原則として有料であるのは，アンバンドリングの考えに基づいている。両者をまとめて提供するバンドリングは，今日でも主にパーソナルコンピュータの分野で，初心者向けにインストール作業の省略，不法な複製を防止する方法として行われている。→(対)バンドル

暗黙の型宣言(default declaration／implicit typing)

プログラムの中にデータのタイプや長さなどの属性を明示的に指定しなかったとき，システム(コンパイラ等)が自動的にデータのタイプ(型)や長さを選択すること。FORTRANの場合，何も宣言しないときには，変数名の最初の1文字で定める。最初の文字がＩからＮの間にあれば整数型と定義し，それ以外の文字の場合は実数型と定義する。暗示的宣言ともいう。

暗黙範囲符(implicit scope terminator)

COBOLの文の終わりに終止符「．」と空白を置くと，先行する未完のすべての文の範囲が暗黙のうちに終了し，一つの完結文になる。この終止符と空白の組を暗黙範囲符という。暗黙範囲符はプログラムの誤りを引き起こしやすいので，なるべく明示範囲符を使うことが望ましい。ただし，段落は完結文の集まりでなければならないので，段落の終わりだけは終止符と空白とで止める。

イ

移行(systems conversion)

従来の業務システムを新規の業務システムに変更，切り換えする工程を広く移行という。移行は，次の4つの領域で考える。

(1)業務の移行(業務手順やルールの変更・稼働後の運用教育など)

(2)システムの移行(ハードウェアや回線の設置など)

(3)データの移行(旧システムから新システムへの切り換えなど)

(4)運用移行(システム運用要員の教育・保守ルールの確定など)

意思決定(decision making)

企業活動の場で発生するさまざまな問題に対して，多くの情報を収集して分析し，複数の解決案の中から最良の実行案を取捨選択すること。

意思決定支援システム(decision making support system：DSS)

意思決定の手段の一部に用いるコンピュータシステムの総称。経済情報を蓄積，分析するデータベース，多変量解析などを用いた予測やシミュレーション，経営指標の算出，株価や経済動向の予測システムなどを含む。この中には，自社のノウハウを組み込んだエキスパートシステムなど，高度なシステムもある。しかし，最終的な意思決定の主体は人間であり，DSSが提供する情報をいかに使いこなすかがポイントとなる。

→(参)経営情報システム，経営戦略情報システム

異常終了(abnormal end)

予定されないシステムの停止。

移植(port／porting)

あるハードウェアやOSのもとで利用できるソフトウェアを手直しして，別のハードウェアやOSで利用できるように修正すること。ソフトウェア開発の効率を高めるために，既存ソフトウェアを活用しようという考え方から生まれた。ソースプログラムをコンパイルし直せば，そのまま別機種で実行できる場合，データ媒体，データが他機種で利用できる場合は「互換性がある」という。→(参)データの互換性

移植性(portability)

ソフトウェアの移植作業のしやすさ。移植性を高めるためにはもとになるソフトウェアの作成段階から次のような配慮が必要である。

(1)標準的な高水準言語を使用する

(2)特定の機種に依存する機能を，限られた数のモジュールに集中する

位置決め時間(positioning time／seek time)

磁気ディスク装置でアクセスアームの位置を適切なトラックの上に設定するために必要な時間。

一時記憶装置(temporary storage)

永久記憶装置に対比した語で，電源を切ると記憶内容を失う記憶装置。一般にはRAMをさす。

一時ファイル(temporary file／transaction file)

永久ファイル(マスタファイル)に対比した語で，短期間使用するファイル。→(例)トランザクションファイル，テンポラリファイル

一括コンパイラ(one-pass compile)

→(参)コンパイラ



一括処理方式(batch processing system)

→(同)バッチ処理方式

一括変換

日本語の入力時，日本語FEPに文章を切れ目なく入力し，一括して変換を行う方法。各文節ごとに変換する方式に比べ，誤った文節に解釈される場合があるが，作文中の考えが中断されないなどの利点もある。また前後の文節や単語を解析し，最適な漢字に変換するFEPも登場している。

一斉テスト(whole test)

小規模なシステムのテスト法で，モジュールテスト(単体テスト)や結合テストを省略して，各モジュールの開発が終わった時点ですべてのモジュールを結合して行うテスト。エラーが存在した場合，原因の特定が困難な手法で，大規模なシステムのテストには適していない。一部を修正しただけのモジュールやごく小規模なシステムのテスト法である。

移動平均法(method of moving averages)

過去のデータに基づいて将来を予測する手法のひとつで，一定期間に区切って期間内の平均値を順次に求め，各移動平均値から季節変動を求める方法。さらに季節変動調整済みデータに回帰曲線を当てはめて予測したいときの推定値を求める。

イベント(event)

マウスのクリックやキーインなど，プログラム実行中に生ずる外部からの介入のこと。

イベント駆動(event driven)

イベントが発生した時点で，対応する処理が実行されるようにプログラム内の制御を書いておくこと。対話型のOSやアプリケーションでは，処理の流れは定型化されておらず，操作中のユーザの要求次第でシステムの対応を変える必要がある。イベント単位で独自に動作する多数の小プログラムを作っておき，システム全体として矛盾なく，スムーズに動作するアプリケーションプログラムの作成方法をイベント駆動型プログラミングという。

意味解析(semantic analysis)

翻訳プログラムの処理は，字句解析と構文解析を行い，その後，意味解析を行う。例えば型の概念のある言語では，変数が整数型，実数型などの型をもつが，各変数がもっている型の意味を解析し，変数名と型の一覧表を作成することなどがこれに相当する。→(参)構文解析，字句解析

イメージ(image)

出力した画像データのこと。基本的に画素で表現する情報。コンピュー

タで扱う写真や図版，イラスト，地図などをさす。

イメージデータ(image data)

画像データのこと。普通，イメージデータは画像をごく小さな領域(ピクセル)に細分化し，白黒の場合そこが白か黒かのいずれか近い方のデータにする。また，カラーの場合は，その領域の色に最も近い色素の組合せのデータになる。細分化の度合いを細かくすればするほど，実際の画像に近く見える。この細分化の度合いを画像の解像度，または画像品質と呼び，dpi(dot per inch)の単位で表す。ただし，解像度を高くするとそのままではデータ量が大きくなるため，イメージデータを圧縮して保管，処理することが多い。→(参)JPEG

イメージプロセッシング(image information processing)

画像・映像を構成する情報を処理すること。画像・映像情報の入出力・加工・処理だけでなく，画像の認識等も含む。医学用のCTスキャンやランドサット，ノア等の人工衛星からの地理情報処理などが実用化されている。画像情報処理ともいう。

入口点(entry point)

プログラム中の特定の場所で，他のプログラムやルーチンから実行の制御が渡される場所のこと。例えば，サブプログラムの先頭，サブルーチンの最初のアドレスなどが入口点になる。入口点に対応する語が出口(exit)，出口点(exit point)で，実行中のプログラムやルーチンから抜け出る場所をさし，その命令の実行後は制御が他に移る。

入れ子(nest／nesting)

ある種の一つまたはいくつかの構造を，同じ種類の構造の中に組み入れること。再帰的なアルゴリズムは入れ子の構造をもっている。またプログラミングでも再帰的な構造が多い。例えば，繰返し処理の中に同一形式の繰返し処理が入ること，サブルーチンがサブルーチンを次々と呼ぶこと，割込み処理プログラムが，さらに優先度の高い割込み要因による割込みを受け，その処理プログラムを起動すること等が入れ子の構造となる。入れ子の内部の処理が終了すると，一つ外側へ制御が戻る。プログラム言語の処理系により，入れ子の記法，限界値が定められている。

インクリメント(increment)

①繰返し処理で変数に加える一定の値。→(同)増分

②変数を1増加させること。例えばC言語では，演算子++で表す。プラスの単位は算術型データでは数字が1，ポインタ型データではアドレスが1データ分増やされる。

インスタンス(instance)

オブジェクト指向プログラミングにおいて，クラス側からオブジェクトを構成員であると見るときの用語。いくつかのオブジェクトをクラスというグループにまとめたとき，あるクラスに属するオブジェクト，またはそのクラスにより生成されるクラスのこと。

オブジェクト指向プログラミングでは，いくつかのクラスを設定し，クラスとそのインスタンスに変数とメソッドを定義する。そしてクラスあるいはインスタンスに関する変数の参照とメソッドの起動は，定められたインタフェースを通してのみ可能である。

インストアマーキング(in-store marking)

商品に付いているバーコードとは別に，小売店が独自に商品に付けるバーコード。

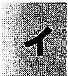
インストラクションミックス(instruction mix)

コンピュータの命令の実行速度を評価する方法。命令ミックスともいう。コンピュータの処理能力を評価するために用いる。命令ごとの実行時間にそれぞれの重み(ウェイト)を掛けた数値の和で表す。和の値が小さい程内部処理能力が高い。それぞれの命令は，使用する頻度によって重み(ウェイト)を付ける。例えば，科学技術計算を意識してウェイト付けを行ったギブソンミックスを表に示す。→(参)ギブソンミックス

命令の種類		重み	命令の種類	重み
データの移動(記憶装置　レジスタ)		31.2	分岐	16.6
			比較	3.8
加減算	固定小数点	6.1	演算(レジスタを用いない)	5.3
	浮動小数点	6.9	けた送り	4.4
乗　算	固定小数点	0.6	論理演算	1.6
	浮動小数点	3.8	指標レジスタ命令	18.0
除　算	固定小数点	0.2	合　計	100.0
	浮動小数点	1.5		

インスペクション(inspection)

もとは品質管理の用語で，製品を検査し，不良品の判定を下すこと。システム開発では，レビューの場での批評をさす。特に，後工程でソフトウェアに欠陥が発見された場合，前工程のどの段階で織り込まれたか調査し，その部分の担当者に説明させ，担当者の誤解や思い違いを明らかにすること。再度の誤りを防止することのほか，技量の向上を図るねらいもあ



る。ソフトウェアの品質向上の手法であると同時にOJTによるエンジニアの教育の場でもある。→(関)レビュー

インタフェース(interface)

①独立に動く二つ以上の構成要素の境界，または境界で共用する部分。二つの異なるシステムを接続し効率よく動作させるために種々の調整を行うサブシステム。装置間でのデータのやりとりを正確に行うハードウェアやソフトウェア，あるいはソフトウェア間における規約など幅広くさす用語。→(関)ヒューマンインタフェース，ユーザ・網インタフェース

②モジュール化したプログラムにおいてプログラム間インタフェース，モジュール間インタフェースとは，あるプログラムが別のプログラムを呼び出して利用する際に，データやパラメタを正しく授受し，他のプログラムやデータ領域に副作用(悪影響)を及ぼさないように定義することである。→(同)プログラム間インタフェース，モジュール間インタフェース

インタフェース用ソフト(interface software)

異なる装置間でのデータのやりとりを正確に行うソフトウェア，あるいはソフトウェア間における規約など。例えば，通信ネットワークを利用して多くのコンピュータと通信可能にするためには，通信規約の調整，データの透過性・互換性が重要な要素になる。通信に関しても次のようなインタフェースが必要である。

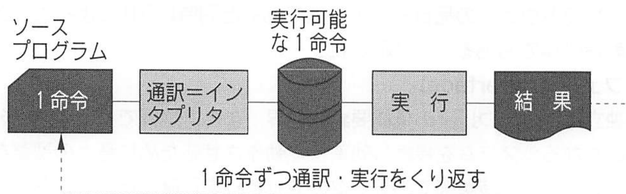
(1)通信インタフェース：コンピュータシステム間のインタフェース

(2)ユーザ・網インタフェース：コンピュータとネットワーク間のインタ

(3)局間インタフェース：ネットワークの交換機間のインタフェース

インタプリタ(interpreter)

プログラム言語で記述したソースプログラムの命令，または中間コードを1命令ごとに機械語に翻訳しながら実行するプログラム。ソースプログラムをまとめて翻訳するコンパイラに対し，解釈プログラム，通訳プログラムともいう。一般に，ひとつの命令に対して複数の機械語の命令をサブルーチンとして対応させている。インタプリタは少ない記憶容量でも動作するが，ループがあるときなどは毎回同じ命令を通訳するので，コンパイラに比べ処理速度が遅い。反面，実行しながらエラーを発見し，その場で直せるというデバッグ上の利点がある。初期のパーソナルコンピュータに多用していたBASICはほとんどがインタプリタ形式であった。インターネットのJavaアプレット(中間コード)を実行するソフトも広義のインタプリタといえる。



インタラクティブ(interactive)

対話型，会話型。キーボードやマウス，ディスプレイ等を介して人間とシステムが情報を交換しながら情報処理を進める方式。典型的な例としては，ワープロ操作，パソコンの表計算ソフトの操作がある。機能の選択，データの入力，結果の出力など，人間の判断と操作を加えながら逐次に処理が進展する点で，あらかじめ処理手順を定めデータファイルを用意するバッチ処理とは異なる。

インタリーブ(interleave)

主記憶装置の複数のバンクを1台のCPUに接続するとき，隣合うアドレスが別々のバンクに割り当てられるようにする方式。先回り制御やマルチプロセッシングに向いている。

インタロック(interlock)

複数の要求に対して同時に応えられない装置で，ひとつの動作中に別の箇所から動作の要求があった時に，後からの要求には応じないようにすること，またはその機構。例えば，複数のタスクがメモリ等の共用資源を使用する場合に，インタロックはタスクの優先順位にしたがって，一時にひとつの要求しか受け付けないようにする。

インディケータ(indicator)

①プログラムによるある処理の結果，指定した特定の条件が生じた場合，所定の状態に設定する機構で，条件コード，状態語，変数の値として示すもの。→(参)条件コード

②ストリームなどのように広がりのあるデータ上の位置を示す変数のこと。指示子ともいう。例えばファイルアクセス中のファイルの位置(ファイルポジション)や，ファイルの終端(EOF)，テキストファイルの行末(NL)などは関数等を用いて調べることができる。

インデックス(index)

①データの所在する位置を示す数字(例えばアドレス)について，別のデータを関連させてこれを変更(修飾)するための値。指標，修飾子ともいう。

例えば，指標レジスタ修飾(インデックスアドレス指定)では，オペラン

ドの示す値にインデックスレジスタの保持する値を加算して，データのある主記憶のアドレスを示す。インデックスレジスタの保持している値をインデックスという。

②ファイル中の特定のレコードや文書中の特定の項目の所在を知るためのキー。索引ともいう。一般には，キーと参照データのテーブルの形式になっている(これは，標題とページが対になって並ぶ目次に相当する)。テーブルはキーの順に並んでいて，特定のデータにアクセスするときは，まずインデックスを探索し，そこから目的とするデータの位置を得てデータをアクセスする。この原理は索引順次編成ファイルで応用する。

→(参)索引順次編成

インデックスコンポーネント(index component)

索引は高速な検索を行うため階層構造をもつ。最下層の索引として，レコードを記録した各CIの情報を管理する順序セットが作られる。順序セットはそれぞれのCIの最後のレコードのキー値が記録されている。順序セットもまた，CIに記録されるが，この順序セットの索引CIの最後のキー値を記録する。これが索引セットである。索引セットはKSDSのレコードが多くなった場合は，索引セットのCIの最後のキーを記録する索引セットを作成する。このようにレコード数が膨大になった場合は複数の階層をもつことがある。

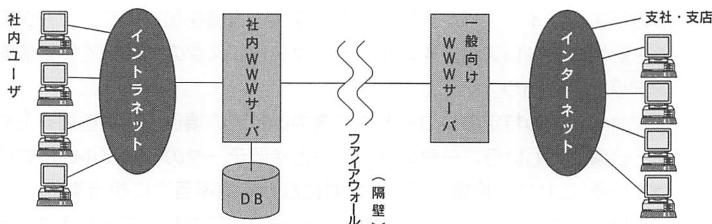
インデント(indent／indention／indentation)

ソースプログラムの記述で，プログラムの構造を視覚的にわかりやすくするために，各行の書き始めの位置を右側にずらすこと。繰返しの構造，分岐の構造，入れ子の構造などを明確にするために，何段階かに字下げする。字下げのためにオートインデント機能(キー)をもったエディタもある。字下げしてもオブジェクトプログラムには影響を及ぼさない。

イントラネット(intranet)

インターネット技術を利用した企業内や学校内の情報システム。WWWサーバを社内ネットワークの中心において業務上の情報を交換する。

社外からのアクセスは一般向けWWWサーバまでにとどめ，社内用サーバには及ばないようにしている。企業で独自のプロトコルやソフトを開発しなくても市場に供給されている経済的で安定した製品が利用でき，統一された仕様で外部との連携も管理も容易である。LANに比べ柔軟性があり，管理も容易で，1995年ころから日米の企業で導入，構築するようになっている。



インハウスデータベース(in-house database)

第三者に販売する目的ではなく，企業などが組織の内部で利用するために作ったデータベース，あるいは，商用データベースから社内利用のためにLANに取り込んだデータベース。企業のインハウスデータベースの保有率は年々増加している。自社の資料センターが所蔵する業務関連の新聞や雑誌，図書などの資料をデータベース化し，利用要求に即応できるようにしたり，商用データベースの頻用の情報を取り込み，専用端末を用いないでも社内LANやイントラネットですぐに使用できるようにしたシステムが多い。

インヘリタンス(inheritance)

→(同)継承

インラインイメージ(inline image)

HTMLでページにイメージデータを表示する場合，実行すると直ちに表示されるイメージデータのこと。比較的データが小さく，多くの人に見せたいイメージはインラインイメージにする。逆に，そのままではタイトルだけが表示され，利用者が見たいとアクション(タイトルをクリック)を起こしたときに表示するイメージデータを外部イメージという。データが大きく，必要な人にだけ見せたいものは外部イメージにする。なお，HTMLのインラインイメージはGIFフォーマットが標準となっているが，JPEGに対応するブラウザも増えている。

ウ

ウィンドウ(Windows)

OSやアプリケーションソフトがディスプレイ画面上に設ける長方形の表示領域。ウィンドウ内は基本的にグラフィック画面であり，文字のほか図形や写真などが表示できる点，その位置や大きさを自由に変えられる点に

特色がある。複数のアプリケーション画面を同時に表示する場合はマルチウィンドウという。

ウィンドウズ(Windows)

アメリカのマイクロソフト社が開発したグラフィカルユーザインタフェースをもったパソコン用基本ソフトウェアのシリーズ名。Windows，Windows 3.1，Windows 95，Windows 98，Windows NTなどがある。

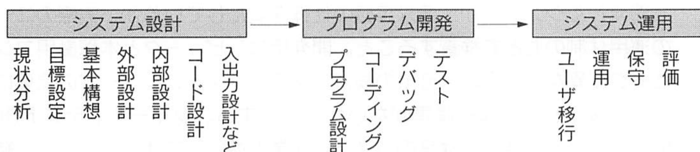
ウォークスルー(walkthrough)

各モジュールのプログラマが同僚の助言を仰ぐことにより，自分が作成したプログラムの仕様や記述の欠陥をさぐること。もとは演劇用語で，役者が監督，演出家に助言を仰ぐ「立ち稽古，リハーサル」の意味である。この場では能力の評価が目的ではないので欠陥を隠す気持ちを起こさせないように管理者を含めないのが通常である。→(参)デバッグ

ウォータフォールモデル(waterfall model)

ソフトウェア開発の諸作業を，基本的に上位工程(上流工程)から下位工程(下流工程)への流れと見なし，各段階では上位工程の成果により作業を進め，下位工程に渡すと考えるモデル。システム規模や特性によって異なるが，一般的には，基本計画(要求分析→要求定義)→システム設計(外部設計→内部設計)→プログラム設計→プログラミング→テスト→運用→保守の段階に区分し，上位から作業を進める。技術的な課題解決は各段階の責任で負う。従来からの主流の開発技法である。

問題点として，上位工程で方針を誤ったりミスを犯したりすると，下位工程の作業が難航したり，矛盾が生じたりする。このような事態では上位にフィードバックして上位工程の設計からやり直す。また最終的な段階にならないと，ユーザが直接に目にするシステムのインタフェース部分の姿が見えないという問題がある。



ウォームスタート(warm start／warm boot)

障害の発生したコンピュータを再起動する際，最初から起動せずメモリの初期化など一部の作業を省略して起動すること。起動時間の短縮を図るだけでなく，障害発生時のメモリの状態を取得することができる場合がある。



受入れテスト(acceptance test)

ハードウェア，ソフトウェアの受け入れにともなう最終の検査。納入側からは引き渡しテストという。要求事項を満たしているかどうかを利用者側のマシンやデータで確認する検査で，設置後，購入側の構内で納入側も参加して行う。→(関)並行テスト

埋込み型SQL(embedded SQL)

既存のコンパイラ言語の中にSQL-DMLを埋め込んでSQLの機能を利用したい場合に使用するタイプのSQL。親言語方式のSQLである。
→(参)埋込み型言語，SQL

埋込み型言語(embedded language)

親言語による適用業務プログラム内の手続きをデータベース操作言語(DML)で記述し，データベース操作を実行でき言語処理系。従来のレコード処理に似た感覚でデータベースのレコード処理ができるようにしたもの。親言語のデータベース処理機能の拡張ともいえる。

上書き保存(overwrite)

記憶媒体にファイルを記録する際，すでにファイルが存在する領域に書き込むこと。もとの内容がなくなるので，注意を要する。

運用(operation)

システムのユーザによるシステムの稼働，システム資源の管理。運用中は，次のような事項を考える必要がある。
(1)性能管理(規定の処理能力を発揮しているか・性能の測定や評価)
(2)障害管理(障害の監視・検知・原因や影響の把握・回復や保守)
(3)運用管理(運用の合理化や自動化・運用経費の算出や課金)
(4)機密管理(システムの情報資源の悪用や盗用の防止)

運用テスト(operations test／field test)

開発したシステムが，要求定義の仕様を満たしているか，実際のユーザの運用体制のもとで検査すること。開発用コンピュータと本稼働用コンピュータが異なる場合など必須の検査である。システムをユーザの環境に移行した後，移行による障害がないかテストする。フィールドテスト(field test)ともいう。通常，従来のシステム作業と並行して運用し，互いの結果を照合しながら行う。

運用日誌(operator's diary)

システムの利用実績を多角的，長期的に監視する目的で残す記録。システムの問題点を早期に明らかにしたり，システム化の効果を判断したりする場合の有効な判断材料になる。障害の記録は特に重要である。問題点の

検討の際には，この他にシステムが自動的に記録したCPU稼働記録，ファイルのアクセス記録，更新記録，周辺機器の稼働記録なども参考にする。

エ

永久ファイル(permanent file)

比較的長期間にわたって保存し，しばしば参照するファイルで，更新しながら継続的に使用する。マスタファイル，基本ファイル，保存ファイルとも呼ぶ。

→(例)マスタファイル，(対)一時ファイル，(参)アーカイブファイル

エイリアス(alias)

①別称，分身の意。システムの複雑な操作を別の簡単な操作に置き換える技法。頻繁に使用するコマンドやコマンド操作を他の短い名前で再定義すること。UNIXやMS-DOSでは，一連のコマンド操作をバッチプログラムに記述しておけば，バッチプログラム名を指定するだけで，同様の操作を行うことができる。

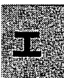
②Windowsなどの操作画面では，階層構造をなすフォルダ内のファイルやプログラムを指定するのに，何回かのアイコン操作に代えて目的のファイルやアプリケーションのアイコンの複製をつごうのよい場所に置くこと。代理のエイリアスを1回操作すれば，システムが階層をたどってアクセスする。そのためのアイコンをエイリアスともいう。近道の意で，ショートカット(short cut)とも呼ぶ。

エクステンション(extension)

ファイル名の一部で，属性を示す文字列。拡張子ともいう。例えば，readme.txtのtxtの部分。

エクステント(extent)

ファイル(データセット)に割当てられたDASDボリューム上の連続した領域，または未割当ての連続した領域。すなわち，磁気ディスクにファイルを記憶させる前に，そのボリュームに確保した書き込みに必要な大きさの領域のこと。確保したエクステントはVTOCに登録しておく必要がある。登録はオペレーティングシステムの仕様にしたがって行う。エクステントが存在する物理的な位置を示す情報をエクステント情報という。



枝(branch)

木構造で，節(node)と節を結ぶ線をいう。枝は，節と節のつながりを示すほか，問題の図式表現としては方向をもって，因果関係や時間的順序を表したり，大きさ(重み／weight)をもって，節間を結ぶ(または移動する)時間や費用を表すこともできる。→(関)木構造

エディタ(editor)

広義には文章，画像，音楽などのデータ作成のために入力，挿入，消去，移動，複写などの編集処理を行うソフトウェア。狭義には，単にエディタといった場合には，一般にソースプログラムや文字データなどの文書を編集するためのテキストエディタをさす。

エミュレーション(emulation)

あるシステムが，別のシステムと同じデータを受け入れ，同じシステムのプログラムを実行し，同じ結果が得られるように，他のシステムを模倣すること。ハードウェアで実現する場合，ソフトウェアで行う場合がある。→(関)エミュレータ，(参)シミュレーション

エミュレータ(emulator)

他のシステムと同じデータを受け入れ，同じプログラムを実行し，同じ結果が得られるように，他のシステムを模倣する装置，またはプログラムのこと。マイクロプログラムを利用している。例えば，パソコンやワークステーションなどのインテリジェントターミナルは，複数のエミュレータを利用することで種類の異なるホストコンピュータの端末として利用することができる。→(関)エミュレーション，(参)シミュレータ

エラー(error)

ハードやソフトの正常でない動作，または好ましくない動作。あるいは，その動作を引き起こした要因をエラーと呼ぶ。動作自体は「故障」と呼んで区別することもある。

エラーチェック(error check)

コンピュータシステムが取扱う情報に誤りがあるかどうかの検査。人間が入力したデータ，伝送されてきたデータ，処理中のデータ，処理結果など情報処理のあらゆる局面でデータの正当性を確認するために行う。

エラーメッセージ(error message)

システムに与えたコマンド，プログラム，データなどに誤りがあって実行不能の場合，システムが返すメッセージ。誤りの可能性がある場合に出されるメッセージは警告(warning)メッセージという。

(1)入出力や操作に関するエラーは，プログラム実行中にオペレーティン

グシステムが知らせる。使用者は，そのメッセージをもとに原因を調べ，必要な修正を行う。ウィンドウズなどのGUI環境では，異常が発生するとダイアログボックスが開き，状況を報告し，応答を要求する。
(2)ソースプログラムを機械語のオブジェクトプログラムに翻訳または通訳する過程で，文法上の誤りがある場合に，その誤りの種類や原因などを表現する短い文章のこと。エラーメッセージが出力された場合には，ソースプログラムを修正して再び翻訳するために，コンピュータに再入力する必要がある。→(類)シンタックスエラー，ワーニングメッセージ

エルゴノミクス(ergonomics)

→(同)人間工学

エレクトロニクス(electronics)

電子技術あるいは電子工学など，電子を制御，利用する技術の総称。トランジスタやIC・LSI・電子管・磁性体等の電子部品に関連する技術や，コンピュータや通信機器，各種家電製品等への応用技術などがある。半導体素子の小型化・高速化を背景にマイクロコンピュータなどが開発され，その応用範囲も一段と広がっている。

エレクトロニックコマース(electronic commerce：EC)

インターネットのような通信ネットワークを利用した商取引。商品やサービスの紹介，受注から決済までの一連の手続き全体を含む形態をさす。
→(参)電子データ交換

エレクトロニックバンキング(electronic banking)

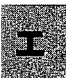
銀行の窓口に行かなくても，企業や家庭内の端末機器から銀行のコンピュータに直接アクセスして振込や振替，残高照会などができる銀行サービス。

演繹的アプローチ(deductive approach)

システム分析を行う際のアプローチ法のひとつで，開発するシステムがまったく新規であったり，既存システムが現状に合いにくい場合，新たなシステムを最初から作る技法。一般に既存技術がないため，理想的な新システムを指向するので時間と費用がかかることが多く，リスクも大きい。目的指向型アプローチともいう。

演算子(operator)

表計算ソフトで数式を定義するとき，または高水準言語でプログラムを書くときに，演算や条件を記述する式の中に使うことのできる演算の動作を表す記号。オペレータともいう。演算子はひとつ(単項)または 二つ(二項)の作用対象(オペランド)をもつ。



エンジニアリングシステム(engineering system)

技術者がCADやCAM，CAE(製品開発のシミュレーションや設計・製図の作業をコンピュータで支援する機能)などを利用し，設計，製造，検査，ソフトウェアの開発，シミュレーションなどを行う。演算処理やグラフィック処理の高性能化を追求したエンジニアリングワークステーションを用い，ネットワークを介して，前工程，後工程と連携し，業務を自動化する。

エンジニアリングワークステーション(engineering workstation：EWS)

演算処理やグラフィック処理の高性能化を追求したワークステーション。ワークステーションの中で，特に技術者がCADやCAM，CAEなどの技術分野で利用する機種で，OSはUNIX系が主流である。一般的にはパソコンより高機能，高価である。

エンティティ(entity)

実体のこと。リレーショナル・データベースを理論的に扱うために必要な概念。

エンドユーザ(end user)

直接的，間接的を問わず，最終的な位置でコンピュータシステムを利用する利用者，利用部門のこと。また，システムの設計，管理，運用等の業務の種類や知識や技術の有無を問わない。端末による対話型処理システムが利用できるようになる以前は，エンドユーザがシステム開発やデータ入力に関わらなかったが，今日では，コンピュータ，アプリケーションプログラム，各種ツールの普及により，エンドユーザ自身が，システムの開発，運用，保守に当たる事例が急増している。

エンドユーザ言語(end user language：EUL)

エンドユーザが容易に扱えるように作られたプログラム言語や対話型のソフトウェア。メニュー，アイコンなどで操作でき，データ処理をマクロ命令などにより指定できることが特徴である。→(参)表計算ソフト

エンドユーザコンピューティング(end user computing：EUC)

情報システムの現実の利用者自身が，自分たちの必要とするシステムを自発的，積極的に構築して利用すること。従来，企業内では必要とするシステムをすべて情報システム部門に開発依頼してきた。しかし，システムの大規模化から情報システム部門は開発要求に対し必ずしもタイムリーに応じることができず，多くのバックログを抱えるようになってしまった。EUCが注目される背景には，このような状況に加え，最近の分散処理システム，ダウンサイジング，使いやすいパッケージソフトウェアの発達などにより，エンドユーザでも開発可能な環境が整ってきたためもある。

エンドユーザシステム(end user system)

基幹システムに対比して，各部門や個人が独自に利用するシステム。

エンドユーザデベロップメント(end user development : EUD)

エンドユーザ自身によるシステム開発や保守作業。通常は，アプリケーションプログラムのマクロ機能やエンドユーザ言語を駆使して，個人や所属部門内の小規模システムを開発したり，既成システムを改造したりすることをさす。アプリケーションパッケージのカスタマイズなどを含む。

オ

オーサリングツール(authoring tool)

文字，グラフィックス，動画，音声などの素材を使ったアプリケーション作成作業を支援するソフトウェア。マルチメディアのアプリケーション作成に欠かせないため，マルチメディア作成ソフトともいう。もとは，プログラミングの知識のない教師がCAI用の教材ソフト(コースウェア)を作成できるように開発されたソフトウェアだった。CD-ROMやCD-Iのソフト作品(タイトル)などビジネス分野での利用も増え，多機能化している。ストーリーの展開に向くもの，情報検索に向くもの，動画の取り込みが簡単にできるものなど，ソフトウェアによって機能に特徴がある。

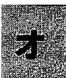
オーダ(order of complexity)

アルゴリズムの評価に用いる計算量を表す方法のひとつで，問題の大きさ(例えば処理するデータ件数，解く方程式の次数などのパラメタ)によって計算量がどう増減するかを示す式。通常はO(オー)記法によって，問題の大きさをnで表したときの増加の度合で表す。

(例)　O(1) ………nが増加しても計算量は不変
　　　O(log n)……nが10倍に増加すると，計算量が2倍になる
　　　O(n) ………nが10倍に増加すると，計算量も10倍になる
　　　O(n log n)…nが10倍に増加すると，計算量が約15倍になる
　　　$O(n^2)$………nが10倍に増加すると，計算量が約100倍になる
　　　$O(2^n)$………nが10倍に増加すると，計算量が約1000倍になる

オートセーブ(auto-save)

アプリケーションソフトに備えられたバックアップ機能のひとつ。操作中のデータを保護するために，ある時点のデータを自動的に補助記憶装置



にファイルとして保存すること。

オートメーション(automation)

自動的手段による作業・処理の遂行およびそのための変更(自動化)。次のようなものがある。

(1)装置工業のプロセスオートメーション

(2)機械工業のメカニカルオートメーション

(3)事務分野のビジネスオートメーション

(1)と(2)をファクトリオートメーション(FA)，(3)をオフィスオートメーション(OA)ともいう。

オーバフロー(overflow)

直接編成ファイルでハッシュ法によりあるレコード(A)のキーからアドレスを算出したとき，そのアドレスがすでに別のレコード(B)に占有されている場合，格納場所のないレコード(A)をオーバフローという。

オーバフロー領域(overflow area)

索引順次編成ファイルにレコードを追加する場合，あらかじめ余分な領域を確保しておいてその場所に書き込む。この場所のことをオーバフロー領域(あふれ域)という。こうすることにより，レコード追加のつどファイルを書き直す必要はなくなる。あふれ域には，各シリンダごとのシリンダあふれ域と，どの場所からでも利用できる独立あふれ域とがある。あふれ域に多数のデータが入った場合は能率が下がるので，ファイル本体を作り直したほうがよい。→(参)索引順次編成，再編成

オーバヘッド(overhead)

OSの制御プログラムが使用するコンピュータ資源。ユーザにとっては，制御プログラムが使用している資源は，自分の処理プログラムで利用できないことから，オーバヘッド(直接生産に関係のない経費の意)という。

通常，OSが使用するCPU時間や記憶領域の大きさをさす。しかし，スループットを向上させるためのオーバヘッドもあるから，オーバヘッドの多少は，必ずしもシステムの全体的な性能を示す指標にはならない。

オープン(open)

①ファイル操作の際，またはプログラム中でファイルを使用するとき，事前に実行する命令。またはその操作。「ファイルを開く」ともいう。ファイル使用にかかわる準備を行う。→(対)クローズ

②メーカが製品の仕様や規格を広く一般に公開すること。

オープンシステム(open systems)

従来は仕様を公開したOSをさしたが，今日では基本機能，操作手順，接

続法などの標準仕様を公開しているコンピュータシステム全般をさす。特定の機種やメーカにこだわらず，仕様さえ合致すれば，異なるメーカの情報機器，通信機器を接続できる環境を提供する。メーカごとにアーキテクチャが異なると，接続できる機種は一定のメーカ製品に限られ，安価で目的にマッチした製品，既得の製品があっても接続できない。

オープンバッチ処理(open batch processing)

コンピュータの操作をユーザ自身が行うバッチ処理方式。ユーザはマシンの起動からデータ媒体のローディング，ジョブの指定，システムの終了までの操作・運用を行う。

オープンプライス(open price)

市場の価格競争原理を尊重し，メーカが標準小売価格を決めないで，流通機構，小売店の決定に任せる販売価格。外観から機能・性能がわかりにくく，新製品の登場が早いパソコンのハードやソフトについては，目安となる標準小売価格があるほうが売りやすく買いやすいという意見と，割引販売やアップグレード価格が存在しては無意味という意見が混在しており，日本のパソコン流通機構では，まだ大勢になっていない。

オフィスオートメーションシステム(office automation system)

オフィス内の情報処理機器を機能的に結びつけ，オフィス業務の合理化，省力化を図り，生産性の向上や経営の改善を目指すシステム。最近では，パソコンを端末とし，プリンタ，電子ファイル，ファクシミリなど，各種OA機器をLANで結び，情報を総合的に管理する統合OAシステムが増えている。→(参)分散データ処理システム

オブジェクト(object)

プログラムやデータを構成する基本的なモジュール。広範囲な内容を含む語で，操作の「対象」となる物すべてを含む。あるオブジェクトは自己の状態をもち，別のオブジェクトと結合し，連絡しあって機能を果たす。メソッドと呼ばれる手続きが，外部からメッセージを受け取り，メッセージと自己の状態から処理を行い，他のオブジェクトにメッセージを渡したり，自己の状態(記憶)を変える。一般には，オブジェクトは小さな処理単位ごとに作り，複雑な処理は，多数のオブジェクト間でメッセージをやりとりしながら行う。

オブジェクト型(object type)

型指定子と型修飾子によって表現されるvoid型以外の型のこと。不完全型をデータオブジェクトタイプに含まないものと，含む考え方がある。含まない考え方を採用すると，オブジェクト型は必ず実体を保持している。

オブジェクト指向(object oriented)

処理の手順(制御構造)とデータ構造を中心とした手続き型手法とは異なり，処理対象(オブジェクト)そのものを中心に展開する情報処理の考え方。WindowsなどのGUI環境は，オブジェクト指向の考え方から生まれた。

データとそれに関する操作や手続き(メソッド)をひとまとめにしてオブジェクトという形式の単位にまとめ，多数のオブジェクト間でメッセージをやりとりしながら全体の処理を進める。処理はオブジェクトへの要求という形で表現する。オブジェクトは他のオブジェクトに関知しないため，オブジェクトの独立性が高まり，プログラムが単純化できる。

→(関)オブジェクト指向言語，オブジェクト指向プログラミング

オブジェクト指向言語(object oriented language)

処理するデータを重視し，プログラム中で積極的にデータを中心に記述できるタイプのプログラム言語。このような言語の例としてSmalltalkやC++，Object Cなどがある。

従来の手続き型言語によるプログラミングでは，もとになるデータ(入力)を，手順を追って処理し，目的の結果(出力)を得るため，プログラミングの中核はデータを処理する手順を考えることにあった。またハードウェアの実行順序もプログラムの記述順に密着している。またデータはファイルという概念でプログラムとは別物と意識することが多い。これに対し，データと手続きに相当するオブジェクトは物，目的，対象であり，処理はオブジェクトにメッセージを送り，オブジェクトは返事(値)を返すという形で進行する。返事は他のオブジェクトへのメッセージになり得る。

オブジェクト指向言語の基本要素は，次の4点である。

(1)カプセル化(encapsulation)　(2)継承(inheritance)
(3)多様性(polymorphism)　(4)均一性(uniformity)

オブジェクト指向プログラミング(object oriented programming)

モジュールの階層化と独立化，部品化を積極的に意識して，オブジェクト指向言語を利用して行うプログラミングの思想，またはその工程，作業。手続き型のプログラム言語によるプログラミングが，データ処理の手順を考えることを中心にしたのに対比し，処理するデータを重視し，プログラム中で積極的にデータを中心に記述できるタイプのプログラミング手法が開発され，オブジェクト指向プログラミングという。

イベント駆動のウィンドウシステムは，オブジェクト指向プログラミングによくあう応用分野であり，オブジェクト指向の考え方はクライアントサーバシステム，分散オペレーティングシステムにも適用されている。

オブジェクトプログラム(object program)

アセンブラ言語やコンパイラ言語で書かれたソースプログラムをアセンブラやコンパイラで翻訳し，生成した機械語コードのプログラム。目的プログラム，ターゲットプログラムともいう。オブジェクトプログラムは，リンケージエディタを介して実行可能にする。→(参)コンパイル

オプション(option)

あるシステムに合致する装置や回路，プログラム，サブシステム，パラメタなどが複数存在し，ユーザが任意のものを選択可能なとき，それらの機器や機能をオプションという。ハードウェアの場合，標準装備されないため別売り品として販売される場合が多い。

オフライン(offline)

周辺装置などがCPUと論理的に直接には結ばれていない状態。通信回線などを利用して得たデータやプログラムをいったん磁気ディスクなどの記憶媒体に記録してからあらためてコンピュータに入力したり，出力するシステム。オンラインシステムの普及以前の一般的なシステム。オンラインに対比してオフラインという。→(対)オンライン

オフラインシステム(offline system)

コンピュータのCPUが端末や周辺装置などと直接には結ばず，通信回線などを利用して送られたデータやプログラムをいったん磁気ディスクなどの記憶媒体に記録してからコンピュータに入力するシステム。オンラインシステムの普及以前の一般的なシステム。

オフライン処理(offline data processing)

オフラインシステムで行うデータ処理。遠隔地で発生したデータを通信回線などを利用して送り，これを受け取った側であらためて媒体に記録してからコンピュータに入力する。データを受け終わってからコンピュータに入力するため，基本的に一括処理になる。

オペランド(operand)

オペレータ(演算子)が作用する対象。単項演算子はひとつ，2項演算子は二つ，3項演算子は3つのオペランドをもつ。→(同)演算子，(同)項

オペレーションズリサーチ(operations research：OR)

経営計算等に用いる数学的手法。企業経営で，使用する人や機械・材料・エネルギー，その他の資源を有効に割り当てて，最大の効果をあげるための科学的計算法などをさす。

オペレータ(operator)

①コンピュータを操作したり，テープを磁気テープ記憶装置にセットしたり，計算結果を取り出すなどの作業を行う人。→(同)システムオペレータ

②表計算ソフトや高水準言語でプログラムや数式を書くときに，計算や条件を記述する式の中に使うことのできる演算の動作を表す記号。

オペレーティングシステム(operating system：OS)

コンピュータ資源，ネットワーク資源を管理し，効率よく働かせるためのソフトウェアの集まり。ユーザのプログラムの実行に必要な種々のサービスや監視を常時行う。このため基本ソフトウェアということもある。システムの処理能力の向上，ヒューマンインタフェースの向上，応答時間の短縮，信頼性・可用性の向上などが，オペレーティングシステムのねらいである。汎用コンピュータ用OSは，各コンピュータメーカが独自のものを開発していたためOS間の互換性はとぼしかったが，ワークステーションやパソコンでは標準OSが出現するようになった。その代表的なものに，MS-DOS，Windows，UNIX，OS／2などがある。また，分散処理システムの普及とともに，NetWare，Windows NTなど通信ネットワークを強く意識した専用OS(ネットワークOS)も現われている。

重み(weight)

複数の要素をもつシステムを評価するとき，各要素の影響力や効果を考えてその要素の値にかける係数。重み係数ともいう。

→(参)ギブソンミックス，インストラクションミックス

親(parent／owner)

①木構造における上位の節。下位に対しては複数のポインタがあり得る。下位の節を子(child)と呼ぶ。

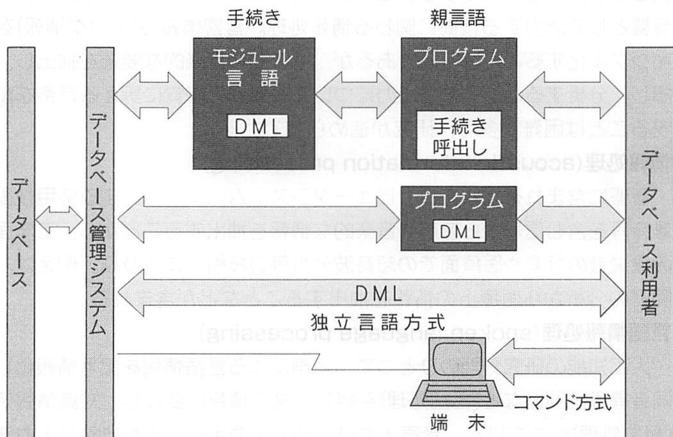
②親子集合型(set type)のデータベースで，相対的に上位のレコードを親レコード，下位のものを子レコードという。この呼び方はシステムで異なることがある。階層型の代表的なDBMSであるIMS(IBM)では，レコードをセグメント(segment)といい，親をペアレント(parent)，子をチャイルド(child)という。また，CODASYLのDBTGでは，親をオーナ(owner)，子をメンバ(member)という。

親言語方式(host language system)

汎用のプログラム言語の拡張機能として，データベース操作言語(DML)を利用できるようにした方式。既存のプログラム言語とともにDMLが使用できるので，例えば，COBOLのプログラムからDBMSを呼び出して，CODASYL方式のデータベースシステムや関係データベースシステムを利

用して，データベースから読み出したデータをCOBOLプログラムで処理した上，再びデータベースに書き戻すことができる。

このように汎用プログラム言語(データベース言語からは親言語という)とデータベース言語を併用する方式を親言語方式という。NDLでは，親言語として，COBOL，FORTRAN，PL／I，Pascalを指定している。親言語方式には，モジュール言語を利用する方式と埋め込み言語を利用する方式がある。



親子集合型(set type)

データやデータセットのネットワーク構造で，一方が他方の従属関係にあるとき，その組を親子集合型という。親子集合型のレコードの集まりを親子集合という。相対的に上位のレコードを親レコード，下位のものを子レコード，最上位のものを根レコード(root record)という。

親ディレクトリ(parent directory)

階層構造(木構造)のディレクトリにおいて，カレントサブディレクトリのひとつ上位のディレクトリ。MS-DOSでは「CD..」で移動できるディレクトリ。→(対)サブディレクトリ，(関)カレントディレクトリ

オリジナル(original)

ファイルやソフトウェアの原作，原型，原典。パソコンユーザの間では次のような場合に使う。

(1)ソフトの人気が高いために類似のソフトが多数開発されたとき，最初の，または最有力のソフトをさす。

(2)購入したソフトウェアパッケージに入っているメーカの媒体。運用す

る媒体，あるいはハードディスクにインストールしたプログラムに対してオリジナルという。

(3)ファイル更新をする際にデータ保存のために確保しておく更新前のファイル。更新後のファイルに対してオリジナルという。

音楽情報処理(music processing)

人工知能の応用分野のひとつで，コンピュータシステムにより音楽を認識したり，作曲したりすること。また文字コードで与えられた音楽情報を音響として出力する技術に関わる情報処理。音響情報(アナログ情報)をディジタル化することは容易であるが，そこから音楽的な要素を抽出し，採譜し，分析すること，また出力についても人間の感覚に訴える音楽を創出することは困難が多く，研究が進められている。

音響情報処理(acoustic information processing)

音響に含まれる情報をコンピュータシステムにより抽出する応用技術。雑音等を含む信号から音声や音楽的な情報を抽出することから，工学面での機械音の計測や医療面での超音波分析等の技術にまで及ぶ。例えば，心臓音の分析から医療上の情報を抽出することなどが含まれる。

音声言語情報処理(spoken language processing)

人工知能の研究分野のひとつで，音声による言語情報を記号情報化し，同音異義語判断など(言語処理)を経て，文字情報に変換し，知識情報処理(概念処理)へつなげる。音声入力は，他の入力手段に比べ自然な入力法で今後の発展が期待されている。また文字コードによる表現を自然な音声出力とする音声合成など(対話処理)を行い，自然言語対話処理システムを目標とする研究分野でもある。

音声合成(speech synthesis)

コンピュータからの情報を音声で出力させるために，データに対応する音声を合成して作ること。代表的な方法に人間の声を文・節・語単位にあらかじめ登録しておき，選択編集する選択合成法がある。さらに音声を分析してその特徴をパラメタ化して圧縮記憶させ編集する分析合成，人工音声を一定の方法で合成する法則合成がある。音声合成技術は，入力データのチェックのための読み合わせやワープロ文書の読み上げ，コンピュータの出力を電話で伝える場合などに有効。NTTの電話番号案内，高速道路の運行状況案内，パソコンではテキストファイルの読み上げプログラムなどで実用化している。→(関)音声応答装置

音声多重放送(sound multiplexing broadcast)

テレビ電波のすきま(垂直帰線消去期間)を利用して，音声情報を送信す

る放送。1978年から定常的に放送。ステレオ放送，2か国語放送，解説放送，緊急放送などに利用している。→(参)文字多重放送

音声通信処理システム(voice communication processing system)

単なる電話の通話サービスだけでなく，音声をディジタル化する，蓄積する，送る(音声メールサービス)，別の時間に(メッセージ通信サービス)送る，必要以外の人には聞かれないようにする(機密保護サービス)，多数の人に同じメッセージを送る(同報サービス)などのサービスを行うシステム。これを商品化したものにNTTの伝言ダイヤルサービスがある。

音声データ(audio data)

人の声，音楽，自然の音などをディジタル化したデータ。ディジタル化により合成・加工・蓄積・圧縮・伝送などがコンピュータにより処理できるようなる。Windows 95ではWAVファイルが標準となっている。アナログデータの処理過程で発生しやすい雑音の混入や音品質の劣化に対して非常に強い性質をもつ。最近のディジタル化技術の発達によって，より自然に近い音声を再現できるようになった。音声データの圧縮はADPCM方式で行うのが一般的である。→(参)WAVEファイル

音声入力装置(voice input device)

音声によりマイクを通して直接コンピュータに単語データの入力を行う装置。特定の人の声をあらかじめ登録しておけば，判断能力が高くなる(特定話者音声入力)。音声認識専用LSIの開発により，少ない誤り率で入力できるようになり，実用化が進んでいる。

音声認識(speech recognition)

コンピュータシステムが人間の音声情報を認識すること。音声入力を可能とするための必須技術である。人間の音声の区切りが不明瞭なこと，個人差，認識対象の増大，外部雑音の除去等の課題が多く，現時点では人間の日常的な会話をコンピュータが理解するには至っていない。音声認識専用LSIの開発により，特定話者の音声は低い誤り率で入力でき，単語認識が可能な水準に達している。レベルになった。→(関)パターン認識

音声メール(voice mail)

マルチメディア通信サービスのひとつで，音声情報を電子メール同様に送るシステム。受信者の画面上に本人宛のメールボックスの内容が分かるようになっており，メッセージを聞くことができる。

音声メールサービス(voice mail service)

人の声を郵便のように配送すること。一般の電話の場合は，相手がいないと受け取れないが，音声メールは相手が聞き取る(取り出す)まで蓄積す

るので，相手のいるいないにかかわらず送ることができる。

オンデマンドサービス(on demand service)

端末からユーザが要求した情報をシステムが即座に提供するサービス。マルチメディアを前提に，レンタルビデオをネットワークを通じて行うビデオオンデマンド，必要なニュースだけをネットワークを通じて提供するニュースオンデマンド，オンラインショッピングなどがオンデマンドサービスに含まれる。システムがあらかじめ決められた情報を繰り返して提供するのではなく，ユーザのその場その場のさまざまな求めに柔軟かつ容易に応じることが，このサービスに欠かせない考え方である。

オンデマンド処理(on demand processing)

要求時点で即時に行う処理。→(参)リアルタイム処理

オンライン(online)

周辺装置や端末がコンピュータに直接電気的に結びついて，データのやりとりができる状態(無線通信も含む)。これに対し，データを一方から他方へ送る途中で人間の操作が必要な状態をオフラインという。→(対)オフライン

オンラインシステム(online system)

ホストコンピュータと遠隔地にある端末装置(リモート端末)とが通信回線で結ばれ，両者間でデータの交換を行うシステム。

オンライン処理(online data processing)

オンラインシステムで，データの収集や処理，交換を行うこと。オンラインシステムの処理形態として，大きく(1)オンラインバッチ処理と(2)データが発生した時点で即座に処理するオンラインリアルタイム処理に分ける。また，その利用形態には，データ収集，メッセージ交換，照会(問合せ)応答，遠隔ジョブ入力，タイムシェアリングなどがある。

オンラインデータベース管理システム(online database management system)

データベースをオンラインで遠隔地から管理するシステム。

オンライントランザクション(online transaction processing：OLTP)

端末から頻繁に発生する処理要求(トランザクション)に対し，ある一定時間内に処理(データベースアクセスを含む)を完了し，応答する処理をオンラインで行うこと。OLTPは非常に信頼性と性能が高いシステムを必要とするため，フォールトトレラント型のコンピュータをFEPに使用して処理するシステムが増えている。

オンラインヘルプ(online help)

プログラムの実行中に，その操作法などを知りたいとき，使用中の画面に呼び出し，表示させる操作マニュアル参照機能。通常，キーボードの

HELPキーやディスプレイ画面内のHELPアイコンで指示したり，不明事項にカーソルをあてマウスの右ボタンをクリックすることなどで参照する仕組みが多い。→(参)オンラインマニュアル

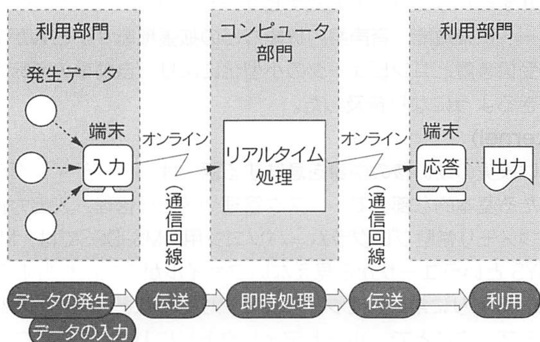
オンラインマニュアル(built-in documents)

電子化マニュアル。操作説明書に相当する情報をシステムの内部に組み込み，利用者が必要になった場合，ヘルプ機能として画面上に提供したり，ガイダンス情報として参照できるようにしたマニュアル。目次検索，キーワード検索などもできる。印刷物として膨大なマニュアルを作成する経費や資源を節約すると同時に，ユーザが必要なつど必要な部分だけを参照できれば十分という考え方に基づいている。なお，初期設定，基本機能，基本操作については印刷物のマニュアルが添付される場合が多い。

オンラインリアルタイム処理(online real time processing)

オンラインリアルタイムシステムで，遠隔地の端末の要求を即時に処理すること。この場合，即時といってもシステムの性格によって基準は異なる。例えば，人を相手とするシステムでは秒の単位，機械を相手とするシステムではミリ秒の単位までというように基準を設ける。

オンラインリアルタイム処理では，端末から対話式で応答を行う方式が多く，代表例には，列車や飛行機などの座席予約システム，お金をやりとりする金融システム，送電線や変電設備の障害を監視する電力系統制御システム，交通量から信号機や道路表示板の制御をする交通管制システム，データベース検索システムなどがある。



カ

カーソル(cursor)

適用業務プログラムからデータベースのデータをアクセスする場合のデータベース操作言語の概念。プログラムとデータベースの間でデータ引き渡しをする抽出条件付きの行である。カーソルを含むプログラムを実行すると，リレーショナルなDBMSにより一時的に結果の表が作られ，この表が順編成ファイルに相当するので，適用業務プログラム側からは，順編成ファイルをオープンしたとみなすことができる。

カード型データベース(card type database)

1レコードのデータを 1枚のカード，またはシートに見立てて記録したファイルを扱うデータベース。単一構造のファイルしか扱わないため，小規模で単純な構造のデータ管理に向いている。いくつかの項目での検索，項目間の計算，条件抽出，集計，分類，並べ換えなどの機能をもつ。リレーショナルデータベースソフトの中には，1レコードを1画面にカード形式で表示する機能をもつものもある。→(参)リレーショナルデータベース

カーナビゲーション(car navigation)

人工衛星の発する電波を受信して自己の位置を正確に知る全地球測位システム(GPS)と地図情報システムを組み合わせて自動車の地理的位置を測定し，運行を支援すること。または，自動車に登載したそのシステム。自動車に登載したジャイロや車速データから算出した自立航法システムの情報を加味するハイブリッド方式ナビゲーション(マップマッチング機能)，自動ルート探索機能，音声案内機能などの拡張機能サービスがある。衛星電波の受信装置とコンピュータの小型化により，自動車の運転席に容易に登載できるようになり普及した。

カーネル(kernel)

もとは果実の硬い核の中身を意味する語。オペレーティングシステムの核にあたる基本的な部分で，タスク管理やメモリ管理，入出力管理などを受け持つメモリ常駐プログラム。パソコン用のMS-DOSでは，MSDOS.SYSと IO.SYS というユーザから見えないファイルがこれに相当する。UNIX系OSでは，メモリに常駐してタイムシェアリングなどの基本的な機能も提供している部分のことで，ルートディレクトリに UNIX というファイル名で保存している。

ガーベイジコレクション(garbage collection)

ガーベイジ(使用済みのデータ)を除去し，現在使用している領域を再編

成して大きな空き領域を確保し，記憶装置の有効利用を図ること。特にLISPやProlog系の言語を使用する場合は不可欠である。→(参)再編成



回帰分析(regression analysis)

2組のデータの間にどのような関係があるか考察する手法のひとつで，説明変数(独立変数)と目的変数(従属変数，被説明変数ともいう)との関係を回帰方程式の形式で示し，分析すること。説明変数の値を指定すると目的変数の数値を得ることができ，予測，推測に用いる。説明変数に掛けられている定数を回帰係数(regression coefficient)という。回帰係数を求めるときには一般に最小二乗法が用いられる。データに回帰方程式を当てはめることを曲線の当てはめ(curve fitting)という。

会計情報システム(business accounting information system)

企業内情報システムのうち，企業会計情報を記録，蓄積，分析，提供する情報システム。取り扱いの対象は，(1)財務的情報，(2)取引データ，(3)前者から得られる経営情報である。

解析木(parse tree)

コンパイラの構文解析過程で，構文要素の解析により作られる木構造の表現。構文解析後の処理に適した構造といえる。構文木とほぼ同義。

階層型データベース(hierarchical database)

データベース内のレコード間の関係が階層型(上位と下位のレコード間の関係を1：Nに限る。木構造モデルともいう)であるデータベース。代表的なDBMSには，IMS(IBM)，DL／I(IBM)，ADM(日立)などがある。

階層型ファイル管理システム(hierarchical file management system)

階層型に設けたディレクトリやフォルダを利用するファイル管理システム。→(参)階層ディレクトリ，ファイル管理システム，(類)HFS

階層構造(hierarchy structure)

広義には，ハードウェア・ソフトウェア・データなどの資源の特性に注目して複数の水準に分け，木構造状に配し，システム全体の効率を高めるように構成したときの構造のこと。各階層はおのおのの役割とする管理・制御の機能をもつ。以下のような例がある。

(1)異なる特性をもつ記憶装置を階層的に配して利用する。

(2)システムの処理機能を分割し，階層的に設計する場合を特に構造化設計という。プログラムの場合，モジュール単位に分割し，階層的に配して構成する場合を特に構造化(構造的)プログラミングという。

→(参)構造化設計，モジュール，トップダウンテスト

(3)データベースやファイルなどで，データやレコード間の特性に注目し

て基本項目(これ以上には細分化できないデータ)をまとめて集団項目とする際に階層的に構成する場合，データは階層構造をもつという。

→(参)データベース，木構造，ファイル管理，階層ディレクトリ

(4)データ通信システムや分散処理システムでハードウェア・ソフトウェアの機能分担を階層的に構成する。→(参)垂直分散構成

階層ディレクトリ(hierarchy directory)

UNIX系OSやMS-DOSなどのオペレーティングシステムが補助記憶装置内のファイル管理の目的で採用している登録簿。WindowsやMacintoshでは，ディレクトリはフォルダの概念と同じ。ファイルの内容，特性の差に注目し階層的に分類，配置ができるように，ファイルを階層的な木構造に位置付けている。→(参)ファイル管理システム

ガイダンス機能(guidance facility)

コンピュータシステムの操作性を高めるために，システムが受け付ける操作や入力方式をメニューやアイコン，アニメなどで示す機能。ディスプレイ画面に常時表示するガイダンスや要求に応じて表示するガイダンスがある。ユーザは，システムを操作するコマンドやパラメタの指定法，入力方式を記憶していなくても指示や選択により操作ができる。またユーザが常時使用するパラメタをシステムが記憶しておき，指定がなければ前回のパラメタに従って処理を実行したり，またパラメタ確認のサブメニューを示す機能もある。

ガイド(guide)

メニューやヘルプとほぼ同義。メニューがコマンド入力の負担を省くのに対し，ガイドは，現在選択されている処理機能やモードを示したり，入力できる文字種や桁数，入力位置などを示すガイダンス情報も含む。

解凍(data expansion／data decompression)

圧縮したファイルをもとへ戻す処理。

概念スキーマ(conceptual schema)

3層スキーマのひとつで，データベースが扱うデータの意味や関係の表現や記述のこと。データベースの応用分野・利用分野の見方や記憶装置に格納する場合の見方とは離れて，人間が認識する意味やデータの構造に近い。この意味で論理スキーマ(logical schema)ともいう。→(参)スキーマ

開発環境(software development environment)

システム開発の各工程でその作業者を支援する環境。開発システムや開発支援ツールなど。→(類)開発システム

開発支援ツール(software development tool)

ソフトウェア開発工程での作業の効率をあげるために利用するプログラム群の総称。→(例)シミュレータ，デバッグツール

開発情報資源(information for system development)

ソフトウェア開発に関する情報を保持するライブラリ。ウォータフォールモデルでは，上流工程で作成された設計書が下流工程で照会され，これに準じて開発作業が進み，プログラムが完成すると，再び上流工程で定義された仕様に合致するかテストが繰り返される。このように開発に関する情報は，関係者のすべてが随時参照できる必要があり，また正しく管理されなければならない。

外部スキーマ(external schema)

3層スキーマのひとつで，データベースの応用分野・利用分野の見地から，データを抽出したり，加工したり，組み合わせたりするための表現や記述のこと。データベースの利用者は，外部スキーマを介してデータベースを利用できる。具体的には，データベースを操作するための記号やデータの指定方法などを含む。→(対)内部スキーマ

外部設計(outline design)

システム設計の領域は，外部設計(概要設計)と内部設計(詳細設計)とに大別できる。外部設計は，システムのおおまかな機能と構成を設計すること。予備設計あるいは基本設計ともいう。現状分析の結果に基づき，その問題点を明らかにしたうえで，要求仕様をまとめ，実現可能なシステムの基本構想をこの段階で作成する。新システムで，どのようなデータを，どう集め(入出力情報の設定)，いかに処理するか(基本機能の概要)，外部システムとのインタフェース，安全性や信頼性のレベルなどを決める。
→(類)システム設計

外部ソート(external sort)

ファイル中のレコードを指定したキー項目について順序よく並べ換えることをソート(整列，分類)という。大量のデータをソートする場合，量に見合う大きな作業エリアが必要であるが，主記憶装置だけでは容量不足となる場合，磁気ディスク装置などの外部記憶装置を用いて分類を行う。これを外部ソート(外部分類)という。

外部表現(external represation of data)

ソースプログラム上の制御文やデータの表現のことで，テキスト形式である。Cでは，通常1バイトの文字が使用される。地域化の指定がある場合は，多バイト文字や拡張文字集合の文字が使用される場合もある。

外部変数(external variable)

関数の外部で定義される変数で，変数の有効範囲が広域的(グローバル)になる。→(対)内部変数，広域変数，局所変数

外部ラベル(external label／sticker)

磁気テープ，磁気ディスク，フロッピーディスクなどの記憶媒体の外側に人間の目で見えるようにつけるラベル。ステッカーともいう。これに対し，記録したファイルを識別するために媒体に電子的に書き込むラベルは内部ラベルという。→(関)ラベル，(対)内部ラベル

外部割込み(external interruption：EI)

計算機システム以外の機器からのコンピュータシステムに対する割込み。次のような場合に発生する。

(1)入出力関係：入出力機器からのデータ発生，入出力機器動作の終了，機器のエラー(用紙切れなど)など
(2)他の機器関係：操作卓からの指示，他のCPUからの連絡など
(3)計時関係：CPU利用の所定時間経過など
(4)誤動作関係：装置内エラー，電源異常など

概要設計(outline design)

→(同)外部設計

改良作業(reform／improve／version up)

ユーザ部門からの機能追加，機能変更などの要求に対処するプログラムの保守作業。運用中のプログラムの一部である場合は，改良で対応するが，システムの根幹にかかわる大幅な仕様の変更になると判断した場合は，新システムの開発に至る場合もある。

カウンタ(counter)

①ソースプログラム中に設けた変数で，1回の処理ごとに定数を加えるもの。繰返しの制御変数や表(テーブル)の添字は代表例といえる。
②データの件数や印字の行数を数える変数。COBOLには特殊レジスタとしてPAGE-COUNTER，LINE-COUTER(予約語)がある。これは，データ部の報告書節中の各報告書記述項に対して作られ，ページや行をカウントするために使用される。

カウントキーデータ形式(count key data architecture)

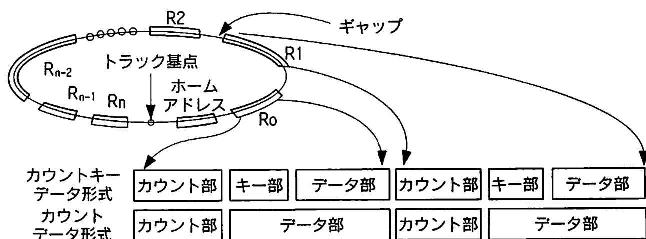
磁気ディスクに索引順次編成ファイルを記録するときの形式。すなわち，カウント部，キー部の後にデータレコード部が置かれる。ギャップがカウントデータ形式にくらべ多くなるため，トラック当たりの記憶容量は小さくなる。

カウントデータ形式(count data architecture)

磁気ディスクに順次編成，直接編成，VSAM編成のファイルを記録するときの一般的な形式。すなわち，カウント部の直後にデータレコード部が置かれる。

カウント部(count area)

磁気ディスク上のレコードが存在するアドレスを示す部分。磁気ディスク上のトラックに記録されるレコード(データレコード部)の前に記録される。アドレス部(address part)ともいう。アドレスを示すシリンダ番号(CC)・ヘッド番号(HH)・レコード番号(R)，キー長，データ長，誤り訂正符号で構成される。



科学技術計算(scientific data processing)

コンピュータの利用分野のひとつである科学技術分野のデータ処理を，特に商業・金融のビジネス分野のデータ処理と対比して呼ぶときの語。一般に科学技術計算は，浮動小数点演算による数値計算が主で，データ量に比して計算量が多いのが特徴である。近年，処理の複雑化・高度化にともなう演算の高速化の要求に応えて，科学技術計算でよく用いる演算形式やデータ構造に合わせた設計によるスーパコンピュータが普及している。

拡張子(extension)

ファイル名の一部であり，ファイル名の後ろに，ファイルの種類や属性などを表したもの。ファイル名は，MS-DOSの場合半角 8文字以内，Windows95の場合256文字以内，ピリオドの後の拡張子は半角 3文字以内。

確率(probability)

ひとつの試行において，ある事象が起こることの確からしさ。ひとつの試行の結果がN通りあり，このおのおのの発生が等確率であるとする。このとき，事象AがN(A)通りの仕方で起こるならば，Aの起こる確率P(A)は，P(A)＝N(A)／N　であり，0≦P(A)≦1である。

カ

確率分布(probability distribution)

確率変数に対する確率の対応のルール。例えば，2つのさいころを同時に振ったときの目の合計とその出る確率で考えると，2つの目の合計は2，3，4，5，6，7，8，9，10，11，12の11種類，これに対応する確率は，1／36，2／36，3／36，4／36，5／36，6／36，5／36，4／36，3／36，2／36，1／36となる。この目の合計を確率変数として，それに対応する確率の分布をつくると確率分布となる。

確率変数(random variable)

ひとつのさいころを振ったとき，1から6までのいずれかの目が出るが，それらの確率は一様に1／6 である。いま，さいころの目をxとすれば，xは1から6までの値をとる変数であり，おのおのの値に1／6という確率が対応している。このように，変数xがとり得るおのおのの値に対して確率が対応しているとき，変数xを確率変数という。とることができる値が離散的であるか，連続的であるかによって，それぞれ，離散形確率変数，連続形確率変数という。

重ね書き(over write)

外部記憶媒体にデータやプログラムをセーブするとき，すでに記録済みのファイルがある記憶領域に，別のデータを書き込むこと。記録媒体に重ね書きすると，前に記録してあったファイルのデータは消滅する。

カスタマイズ(customize)

多数のユーザ向けに設定されているハードウェアやソフトウェアの機能を，特定のユーザの運用形態や使用目的に合わせて変更すること。次のような場合がある。

(1)ハードウェアでは，既製のICの回路を変更して特殊なICを製造すること，企業の大量注文に応じてメモリや電源などを増設したパソコンを製造すること。

(2)ソフトウェアでは，入力時のキーの役割や出力画面のデザインを可能な候補から選択すること，給与計算プログラムや在庫管理プログラムなどのアプリケーションを自社の制度やシステムに合わせて仕様変更することまでを含む。

画像圧縮(image compression)

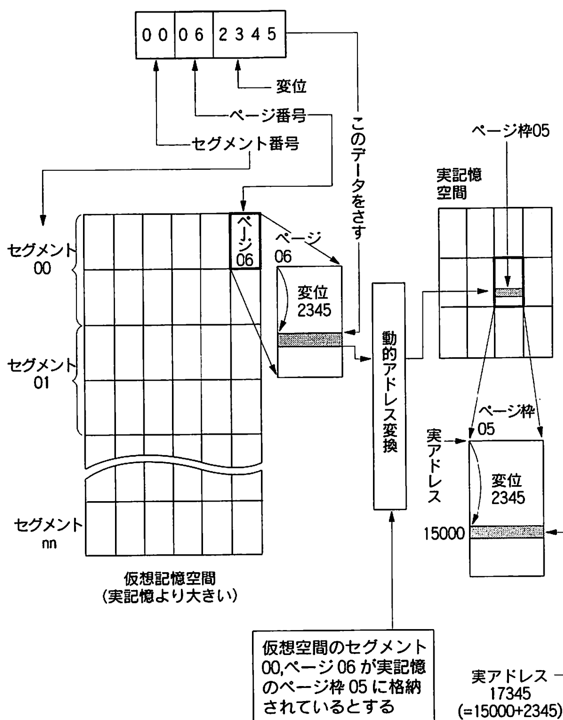
画像を表す情報のサイズを小さくする処理。画像は，コンピュータの処理能力，通信回線の伝送能力に対して，膨大な情報量をもつことが多く，能力に見合う程度の情報量に変換する必要がある。現在のディジタルネットワークの能力では，1秒間の動画に必要な30画面の静止画情報を，1秒

間にリアルタイムで伝送，処理することは非常に困難である。画質の劣化を最小限にとどめ，情報量を減らす技術が必要となる。画像情報では同一色が連続していることや，動画情報では次画面との差分に注目して画面情報を圧縮する技術を使っている。→(参)JPEG，MPEG

仮想アドレス(virtual address)

コンピュータの主記憶装置の実記憶容量以上の大きなプログラムを作成する際の，実アドレスとは別に便宜的に想定した記憶装置上のアドレス。仮想記憶システムにおいてプログラムを実行する際，実記憶装置のアドレス(実効アドレス)に変換される。

仮想記憶(virtual storage：VS)



物理的な主記憶の限界を越え，より大きな論理的な主記憶空間を作り出す方式。主記憶装置と補助記憶装置の間の連携をとって，多量のデータや大きなプログラム，多重プログラミングのためにに複数のプログラムを処理

する技法。プログラムやデータをページという単位に分割し，補助記憶装置にいれておき，当面必要なページだけを主記憶装置にロードし実行する。必要なページが主記憶にない場合は，不要なページを補助記憶に戻し，あらためて必要プログラムをロードし実行する。この方式が可能であるためには，プログラムはあらかじめ主記憶装置のどこにロードされても実行可能なように(リロケータブルに)作っておかなければならない。仮想記憶に対比してハード的な主記憶装置を実記憶装置という。仮想記憶方式により，主記憶装置の記憶容量より大きなプログラムも実行できるようになった。

仮想記憶オペレーティングシステム(virtual storage operating system)

仮想記憶システムを管理する「仮想記憶制御」の機能をもったオペレーティングシステム。例えば，MVS／ESA(IBM)，OS_a／MSP(富士通)，VOS3／AS(日立)などがその代表格である。VSAM(virtual storage access method)編成のファイルは仮想記憶OSのもとで稼働する。

仮想記憶空間(virtual storage)

仮想記憶システムの論理的な記憶空間。論理記憶空間ともいう。仮想記憶空間に対比して主記憶装置の物理的な記憶領域を実記憶空間という。

仮想計算機(virtual machine：VM)

オペレーティングシステムの機能によってハードウェア資源を厳密に管理し，複数のユーザの同時利用，複数のプログラムの同時実行，実在の記憶域以上の記憶域の利用などが可能になった。一般に各ユーザが自分の場で要求するコンピュータ機能・構成は，それぞれ異なっており，物理的に限られたコンピュータ資源を時間的あるいは空間的に分割し再構成することによって，あたかも各ユーザに複数個の論理的なコンピュータ機能を提供できる。このように提供するコンピュータ機能は，ユーザから見れば概念的・論理的に存在するものであることから仮想計算機と呼ぶ。これに対して実在するハードウェア資源を実計算機(real machine)と呼ぶ。仮想計算機はユーザにとって実計算機と同等と見なせるものであるが，実計算機の機能の一部を利用していることになる。

画像情報処理(image information processing)

→(同)イメージプロセッシング

画像ツール(graphic tools)

静止画像を合成・加工するソフトウェア。オーサリングツールの一機能。Photoshop(Mac／Win)，Graphic Converter(Mac)，Paint Shop Pro(Win)などの画像加工専用ツールがある。

仮想ドライブ(virtual drive)

あるディレクトリに割り当てられた仮想的ドライブ名のこと。階層の深い位置にあるディレクトリを指定するとき，ディレクトリ名の入力やパスをたどる操作が増えるわずらわしさがある。仮想ドライブ名を定義し，仮想ドライブ名を入力すれば，そのディレクトリを指定したことになる。

画像入力装置(image input device)

写真や絵画などイメージデータを入力する装置。その原理や機構はさまざまである。パソコンでは，イメージスキャナ，デジタルカメラ，ビデオキャプチャなどをよく使っている。他に，レンズを介して撮像素子からの信号をA／Dコンバータにより変換するテレビカメラ方式，CRT上に対象物をあて管面上の輝点(スポット)を走査(スキャン)しその透過光あるいは反射光をA／D変換するフライングスポットスキャナ，写真や図面などの資料を回転ドラムに巻きつけてヘッドを移動させたり，あるいは逆に資料を平面上に置いてヘッドを移動させるメカニカルスキャナなどがある。

型(type／data type)

データや関数の性質および内部表現。数値データ型としては整数型，実数型，複素数型などの種類がある。実数はその精度により，単精度実数型，倍精度実数型，多倍精度実数型などがある。

型変換(data type conversion／type conversion)

プログラム実行中にデータの型を別の型に変換すること。データ変換ともいう。次の2つの場合がある。

(1)明示的な型変換：組込み関数やキャスト式で行う

(2)暗黙の型変換：代入式や算術式の演算の際に式の右辺の型に合わせて，左辺の型が変わる。

カタログ(catalog／catalogue)

磁気テープや磁気ディスク上のファイルに関する情報(ファイル名やファイル編成，レコード長，記憶した先頭アドレス，参照情報，保護情報など)を一括して記録をつくり，記録によって一元的に管理する場合に，この記録のことをカタログとよぶ。区分編成ではディレクトリ，索引順次編成では索引がこれに相当する。ファイル操作を行うプログラムを実行する際に，OSに与えておく情報である。→(参)ディレクトリ，索引

稼働率(availability／operating ratio)

システムの信頼性，可用性を表す指標のひとつ。システムや機器を利用しようとする時間に対する，システムの稼働時間の比率。1に近いほど，信頼性が高い。システムに故障が発生してから修復までに要する平均修復

時間(MTTR)と無故障で稼働する平均故障間隔(MTBF)との関係から，次式で求める。→(類)可用性

稼働率＝MTBF／(MTBF＋MTTR)

カ

カフェテリア(cafeteria)

本来の意味は自分で料理を運ぶ簡易食堂。情報処理では，クローズドバッチ処理とオープンバッチ処理の中間にあって，ユーザ自身がバッチの登録を行い，処理結果を受け取るが，コンピュータの操作は専任のオペレータが行う方式。オペレータの仕事の一部をユーザに開放した方式といえる。

カプセル化(encapsulation)

広義には，ある単位のデータなり操作なりをまとめて一つのモジュールとし，外部との間に厳密なインタフェースを設けることによって他のモジュールと分離させ，モジュール間の依存を最小にする方法。構造化プログラミングの長所であるモジュール性をより拡張，強化することといえる。

またオブジェクト指向言語では，データ型とそれらに対する手続き(オブジェクト)が，クラスの中に統合され，外部のクラスに対しては内部の細かい作りを見せる必要がなく，必要な情報だけを与えれば用が足りること。

可変長レコード(variable length record)

ファイル中の論理レコードの長さがそれぞれ異なるレコード形式。表計算ソフトやデータベースソフト，通信ソフトでは，決められた長さ以下なら，可変長レコードでも意識したり，指定する必要はない。

常に一定の長さ以下のレコードを扱う場合には，そのハンドリングの容易さから固定長レコードを多用する。しかし，各レコードの長さが著しく異なる場合は，固定長レコードでは無駄が多くなるため，スペース効率の観点から可変長レコードをもつファイルを使用する。システムは，レコードの先頭にそのレコード長を，ブロックにまとめた場合にはブロックの先頭にブロック長を記録する。データの長さを記録する場所は，データの場所を示すカウント部の内部になる。

可変長レコードファイル(variable length record file)

可変長レコードで構成されるファイル。磁気ディスク上にレコードを記録するとき，各レコードの先頭にレコード長を示す領域を有し，また，ブロックの先頭にもそのブロック長を記述する領域をもつ。これらは，データの場所を示すカウント部に記録される。索引順次編成の場合は，キー部を設け，アドレスに代えレコード内のキー項目で特定のレコードを探すことができる。

画面設計(screen design)

入出力設計の工程で，ユーザ端末を使用して入出力する画面を設計すること。画面設計は，ユーザの利用局面が変わると，画面形式も変わるのが普通である。

(1)会話型(対話型)処理：メニューやアイコンの利用，操作の確認

(2)問合せ処理やデータ配信：ダウンロードや印字への配慮

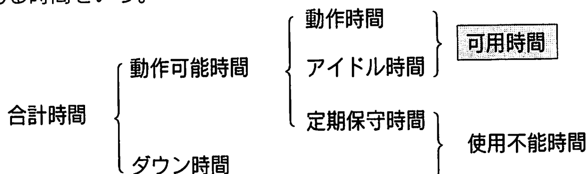
(3)データ収集：操作性，誤入力チェック，メッセージ，再入力の要求

画面レイアウト(screen layout)

画面に表示するデータ項目の並び，レイアウトを決定する工程。画面サイズを考慮し，ユーザに操作しやすく設計する。特に，入力操作をともなう画面は，誤入力を誘うことがないように，理解しやすく，扱いやすくする配慮が必要である。表示したい内容が画面枠より大きい場合は，スクロールやページ分割などの手法を用いる。

可用時間(available time)

利用者の立場からみて，コンピュータの機能単位を使用できる時間。コンピュータが実際に使われているかどうかに関係なく，使用が可能な状態にある時間をいう。



→(参)稼働率

可用性(availability)

システム，装置などが正常に動作している時間の割合。システムの故障が少ないこと，自動回復機能が備わっていること，修理時間が短いことなどが可用性を高めることになる。可用性を表す指標として稼働率がある。

→(類)稼働率

カラムインディケータ(column indicator)

ワープロソフトの入力時のカーソルの所在位置を示すために，画面の上部，または左側に表示する目盛り，または数値。それぞれカラムゲージ，ラインゲージともいう。グラフィックソフトの場合はルーラ(ruler)という。

カレントウィンドウ(current window)

ある時点でユーザの直接的操作の対象となっているウィンドウ。

→(同)アクティブウィンドウ

カ

カレントディレクトリ(current directory)

ファイル操作のためにユーザが選んだディレクトリ。パスを移動して，ファイル操作が可能になっているディレクトリがカレントディレクトリである。UNIX系OSやMS-DOSの階層的ファイルシステム(FMS)では，起動時点で一般的にルートディレクトリ(ホームディレクトリ)に位置づけられる。ユーザは，自分に必要なファイルの所在する階層的ファイルシステム中のディレクトリを移動してアクセスできる。

→(関)サブディレクトリ，(参)パス

カレントドライブ(current drive)

ワークステーションやパーソナルコンピュータに接続されている複数台のディスクドライブのうち，現在のファイル操作の対象になっているドライブ。ボリューム指定がされていない場合は，カレントドライブがアクセス対象となる。

環境(environment)

①広義には，コンピュータシステムのユーザに与えられるヒューマンインタフェースや資源。例えば，プログラミング環境，開発環境，作業環境，利用環境，GUI環境，OSI環境などと広い意味で使用する。

②システムのハードウェア，ソフトウェアの構成。ハードについての環境とは，周辺装置を含めた機器構成を意味する。ソフトについての環境とは，パラメタや変数の値，それらの保持関係などを意味する。

→(参)プログラミング環境，開発環境

関係(relation)

リレーショナルデータベースの基本となるデータ構造で，同じ属性集合の組であるタプル(tuple)を集めた集合。関係表の行に相当するタプル1組は，属性集合の数だけ値をもつ。関係は行の集合であり，2次元の表で表される。この意味で表(table)と同義である。

関係代数(relation algebra)

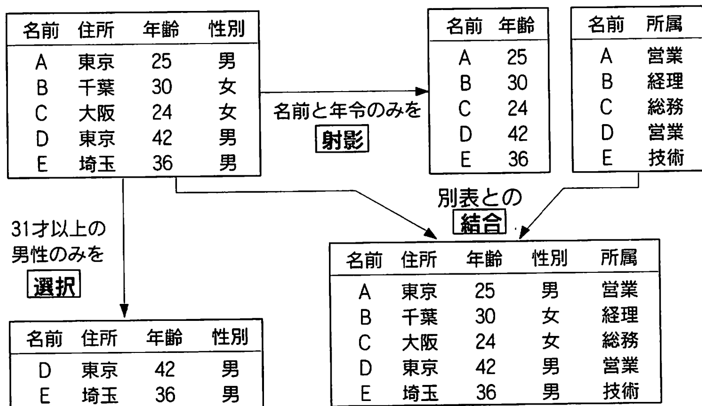
リレーショナルデータベースの表に対する操作を定義する演算系の集まり。この演算を適用することによって，必要な情報を自在に取り出すことができる。E.F.Coddは，選択，射影，結合，除算の4つの関係演算と従来の集合論にあった4つの集合演算(直積，合併，共通，差)を提案した。これらを整理して，今日のリレーショナルデータベースが使用するデータ操作言語になった。次の3種類の基本演算がある。

(1)選択(selection)：表の中から指定の条件を満たす行のみを取り出す

(2)射影(projection)：指定した列のみを取り出す

(3)結合(join)：共通するデータの値によって複数の表から新しい表を作り出す

代表的なDBMSには，SQL，DB2(IBM)，ADABAS(Software Age)などがある。



関係データベース(relational database：RDB)

関連するデータを，行と列に展開した二次元の表(テーブル)形式で取り扱うデータベース。リレーショナルデータベースともいう。IBM社のE.F.Coddが提案したデータモデルに基づく。人間が日常的に使う表の形式にイメージが近く，わかりやすく使いやすい点が特徴である。表と表との関係づけを，データの値によって行う。データ構造表現が柔軟で，データの独立性が高い。表のデータを自由に取り出して加工でき，設計や拡張，変更が容易である。表の操作は集合論の関係代数を利用して演算で行う。

関係モデル(relation model)

データ間の関係を配列や二分木といった特定の実現方法によらず，やや抽象的なレベルで関係づけしたもの。データの集合を関係表と呼ばれる表形式でまとめた関係データデータベースなどがある。

関数(function)

①ある変数の値が，他のいくつかの変数の値を与えたとき，一意に決まる関係にあるとき，この関係を関数という。数学では，例えば，x，yの値でzが定義できるとき，$z=f(x, y)$のように表現する。

②ある引数値を与えると，あらかじめ定義してある手順に沿って計算など

の処理を行い，その結果値を返す機能。プログラム言語や表計算ソフトなどで使用する。例えば，SUMという関数にいくつかの値を与えると，その合計値を返す。

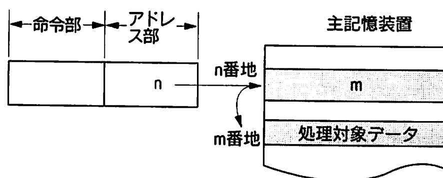
関数型言語(functional language)

関数にパラメタのデータを与えると，値が求められる。関数を合成して新たな関数をつくることができる性質に基づいて，問題を解決する方法がある。関数を式で表し，式の入れ子構造でプログラムを表現する言語を関数型言語という。プログラミングは関数を組み合わせて目的の関数をつくる。LISP，APLなどが代表的な関数型言語である。

→(参)手続き型言語，オブジェクト指向言語

間接アドレス指定(indirect address／indirect addressing)

アドレス指定の一方式。命令語のアドレス部で，処理対象データを格納する物理アドレスを記憶する場所を指定する方式。物理アドレスを記憶する場所は，レジスタや主記憶装置の別の場所である。



間接アドレス方式(indirect addressing)

直接編成ファイルでレコードを指定する一方法。レコードのキー値をそのまま，シリンダ番号(CC)，ヘッド番号(HH)，レコード番号(R)に当てて直接的にアドレスを指定する直接アドレス方式に対比する語。直接アドレス方式は，キーに欠番が多い，レコードのキー値が大きすぎるなどの場合は不向きである。そこでレコードのキー値にある演算を施し，その結果を利用してレコードのアドレスとする間接アドレス方式を用いる。ここで，キー値をアドレスに変えることをアドレス変換(address translation)またはランダマイジング(randomizing)という。

どのような演算を施すかはキー値の分布によるが，除算法・数字分析法・基数変換法，重ね合わせ法などで，アドレスが均一に，かつ無駄なく分布するようくふうする。しかし，あるレコードのキー値に演算を施した結果が別のキーのレコードアドレスと同じになることがある。その場合これらのレコードをシノニムレコードという。

→(対)直接アドレス方式，(関)シノニム

管理情報システム(management information system)

企業内情報システムのうち，企業内部の組織，財産，技術，人事などの情報を収集，蓄積，分析する情報システム。企業の計画，行動，成果などについての指揮，管理，統制に必要なコミュニケーションシステムを含む。

管理台帳(resources manage ledger)

多岐にわたるシステム資源を管理するために作成する台帳で，変更，拡張，移動，交換，バージョンアップのつど，書き換えて最新の環境，状況を正確に把握できるようにする。必要ならファイルの容量，ディスクの空き容量なども管理する。

キ

木(tree)

木構造の略称，または木構造をもつデータの集合。→(参)木構造

キー(key)

①データ項目中に含む1個以上の文字であって，そのデータを認識したり，その用途を制御したりするために使えるもの，または，レコードを検索したり，整列したりする場合の手がかりとなるデータ。データコードやテーブルの指標など。→(参)インデックス，添字，ソートキー

②データを暗号化し，または暗号文からもとの文を復元する(復号する)ために必要な文字列，または数値。暗号化かぎ，復号かぎのこと。

→(同)暗号化かぎ

キー項目(key item)

データの検索や並べ変えあるいは集計などの際のよりどころになる項目。例えば，商品別売上日報を作成する際の商品コードや，個人別成績表を作成する際の出席番号がこれにあたる。

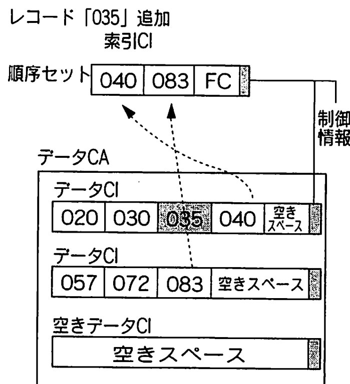
キー順データセット(keyed sequenced data set：KSDS)

索引編成ファイルと同様に，順アクセスと索引による直接アクセスが可能なデータセット。KSDSは索引を作成するインデックスコンポーネント(索引構成要素)と，レコードを記憶するデータコンポーネント(データ構成要素)からなる。

KSDSに対するアクセスは，順アクセス，直接アクセス，動的アクセスが可能である。直接アクセスの場合，アクセスするキー値を指定すると，

索引セットから順序セットを特定し，さらに順序セットからレコードのあるCIを特定する。さらにCIを探索して目的のレコードにアクセスする。

KSDSに対しては，更新・追加・削除の処理が可能である。レコードの追加の場合は，空きスペースを使って，キー値の示す位置に追加される。例えば右図でキー値が035というレコードを追加する場合は，030のレコードの次に追加され，040のレコードは035のレコードの後に移動する。また，追加・削除により，CI中の最後のレコードのキー値が変わった場合，索引の内容も更新される。



キー部(key area)

索引順次編成ファイルのレコードを記録する場合，カウント部に続けて設けるキーを記録する部分。ディスクアドレスによらず，キー項目で指定レコードを早く探す。キー部は，キーと誤り訂正符号で構成される。

キーワード(key word)

①表題や文献の中にあって，その文献の内容を表現する有意な手がかり的な役割を果たす語。情報検索において，情報を類別し，または索引を作るために用いる。→(参)シソーラス

②パラメタ(ある目的または処理のために定数値を与えられる変数)を識別する記号。

③指令オペランドの一部で，特別な文字からなるもの。

記憶スキーマ(storage schema)

データベースの構造と内容の記述をスキーマ(schema)という。スキーマに記述された内容を記憶装置に実現する役割を果たす。ANSIの3層スキーマの内部スキーマに対応する。

CODASYLのDBTGでは，データベースの論理構造をスキーマとサブスキーマの2階層でとらえているが，さらにそれらのスキーマに記述された内容を計算機上にどのように実現するかを記憶スキーマで記述する。

機械語(machine language)

コンピュータが直接に実行可能な命令コードの集合，または命令コードで書かれたプログラム。2進コードで表現する。事実上，人間が高水準言語で書いたソースプログラムをアセンブル，またはコンパイルした結果として出力する命令コード(オブジェクトプログラム)である。この形式は，コンピュータの機種ごとに異なり，相互の互換性がない。

機械翻訳(machine translation)

コンピュータを使って日本語や英語などを相互に翻訳すること。人工知能研究の一分野として始まった。システムの高速化，大容量化により実用化に至ったが，両言語に通じた人間の翻訳者の水準には至っていない。翻訳の方法を大別すると，文章配列が似ている言語間に適用し，コンピュータに内蔵した2か国語の辞書から訳語を当てはめていく構文解析と呼ぶ方式と，文章配列の異なる言語間に適用し，人間には理解できない中間語に翻訳し，さらに目的の言語に翻訳する意味解析と呼ぶ方式がある。

機械向き言語(machine oriented language)

機械固有の機械語に1対1に対応する記号命令語を主とするプログラム言語。機械向き言語には，操作コードに英数字の略記号を使い，アドレス指定にも記号番地を使えるようにした記号言語が多い。アセンブラ言語が代表的。FORTRAN，COBOLなどの高水準言語に対比される。

規格(standard)

工業製品の要素の品質や用語表現，計測法などについて国や標準化団体が定めた標準。規格を定めることで標準化が進み，部品，製品，プログラム，データなどの互換性を保つことができる。情報処理の分野では高水準のプログラム言語で書かれたプログラムは，機種が変わってもほとんど修正することなく使用でき，優れた移植性を確保できる。ハードウェア，ソフトウェア，通信規約について，国内のJIS規格，国際的なISO規格，IEC規格など多くの規格がある。→(参)標準化，デファクトスタンダード

基幹システム(backbone system／corporate system／enterprise system)

全社的な情報システムの中核をなす主要なコンピュータシステム。部門別，個人別の小規模なシステムに対比して，会社の基幹業務に関わる大規模なシステムをさす。

キ

基幹データベース(backbone database／enterprise database)

全社的な情報システムの中核をなすデータベース。外部に提供する商用データベースと異なり，自社内のユーザに限り使用できる場合が多い。

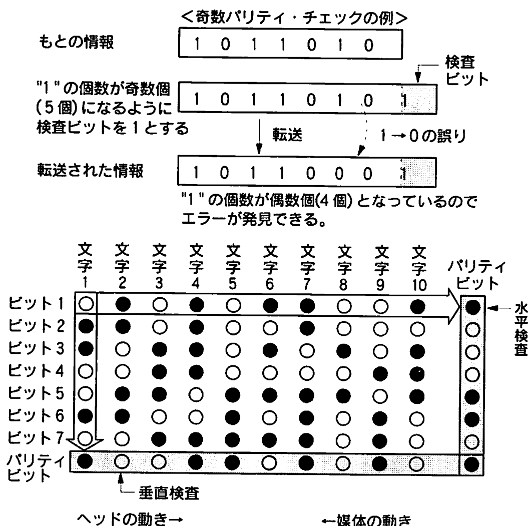
奇偶検査(parity check：PC／odd even check)

情報を転送する際，誤りなく情報が送られたかどうかを，コードの桁数字の和に注目してハードウェア的に検出するチェック方法のひとつ。情報のまとまりごとに1ビットの検査ビットを設け，そのビットを含め1であるビットの総数が常に偶数個(または奇数個)になるようして転送する。転送中に1ビットの誤りが発生するとビットの和の奇偶が合わなくなるため，エラーを検出できる。

(1)1であるビットの総和が常に偶数となるようにする方式を偶数パリティチェックといい，奇数にする方式を奇数パリティチェックという。

(2)2進数字が行列の形式になっているときに，行に対する奇偶検査を水平パリティチェック，列に対する検査を垂直パリティチェックという。

(3)磁気テープなどの媒体に記憶した 2進コードの検査の場合は，媒体の運動方向に対し平行な方向のビットについて行う検査を水平検査，垂直な方向のビットについて行う検査を垂直検査という。



木構造(tree structure)

グラフ理論においては，環や閉路をもたないグラフを木(tree)と呼ぶが，情報科学においては，データなどの要素の上下関係を含めて議論することが多い。情報処理の分野では，構造化プログラミング，データベースのデータ構造，データの検索，ソート，コンパイラの構文の解析などと関連する重要な概念である。点を連結する線が木の幹から枝が出，枝から小枝が出るように分岐を繰り返して構成したグラフの形態をさす。上位に対しては唯一，下位に対しては複数のポインタがある構造を木構造という。上位の点を親(parent)，下位の点を子(child)と呼ぶこともある。木構造は，いくつかの型に分けられる。

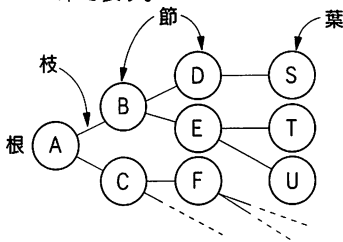
(1)2分木(binary tree)：根(root)側から葉(leaf)への道をたどるとき，節から二つ以下の枝が出るもの

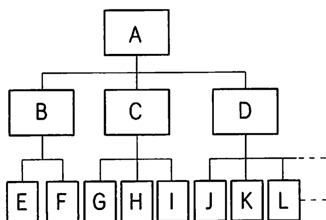
(2)線形木(linear tree)：枝が1本しかないもので，配列やリストの線形構造に相当する

(3)多分木(multiway tree)：各頂点から最大m個(m≧3)の枝が出るもので，2分木と区別する

(4)順序木(ordered tree)：一つの頂点が子をいくつももち，子の間に順序があるもの

(5)有向木(oriented tree)：枝に方向性をもたせ，意味を与えたもので，矢印で表す。





記号論理学(symbolic logic)

自然言語のもつあいまいさや論理的不十分さを避けるために，記号表記法や数学的概念を用いて正しく論証を行う学問。

机上デバッグ(desk debugging)

作成したプログラムを実際にコンピュータでコンパイルし，データを入力してテストする前に，コーディングと流れ図をもとにプログラムの論理を目で見て検査し，誤りを検出して手直しすること。デバッグは，プログラムの誤りを検出して，手直しすること。→(関)デバッグ

期待値(expectation，expected value)

確率変数 x のとる値に x の発生確率P(x)を掛けることによって得られる重みつき平均値。離散分布に対しては，期待値　$E(x)=\Sigma x_i P(x_i)$，確率密度関数f(x)をもつ連続分布に対しては，

$E(x)=\int_{-\infty}^{+\infty} x f(x)dx$　となる。→(参)確率

機能設計(function design)

外部設計(概要設計)で，システムの機能をサブシステム単位に分割し，その内容と関連を明確にすること。→(類)機能分割

帰納的アプローチ(inductive approach)

システム分析を行う際のアプローチ法のひとつで，現状の具体的な問題点の原因を探り，これを解消する方向で対応する技法。既存システムに新機能を付加する，一部を修正して要求に対応する，不要機能を削除する場合が多い。現状に大幅な改変を加えない点で，小さな改善，緊急性の高い問題に短期間で対応する場合によく用いる。

機能テスト(functional test)

内部構造より機能に重点をおいて行うテスト。システム，モジュール，プログラムのテストの際に行う。仕様書に基づいて，システムが実現すべき機能をもれなくテストできるようなテストデータを用意して実施する。

機能分割(function division)

システムの機能をサブシステム単位に分割すること。機能設計の段階で行う。具体的には，次のような作業(ドキュメント作成を含む)を行う。

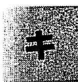
(1)サブシステムの構造と関連(モジュールとインタフェース)
(2)プログラムの外部仕様(機能，入出力仕様等)
(3)システム処理方式(入出力方式，サポート機能等)
(4)障害対策(リカバリ範囲，チェックポイント等)
(5)入出力設計(画面，帳票，メッセージ等)
(6)論理データ設計(ファイル，データベース，データの論理構造等)

ギブソンミックス(Gibson mix)

科学技術計算でコンピュータの計算能力を評価する目的で，J.C.Gibsonが考案した数値。計算処理を行う際の固定小数点加減算，浮動小数点加減算，ストア，判断などの個々の命令の，実際のプログラムでの使用頻度を調べ，対象のコンピュータの命令の実行速度をそれぞれに掛け合わせて合計して求める。小さい値ほど処理速度の速いマシンといえる。1つ命令の平均実行時間を知るのに便利である。
→(参)インストラクションミックス

基本計画(basic plan／primitive planning)

システム開発の最初の段階で行う，システム概要と開発工程の計画。大きく分けて3つの要素からなる。



(1)システム化計画：現状を分析し，問題点を明らかにし，解決策をシステム化計画として立案する

(2)プロジェクト実行計画：システム化計画に沿って，概要を設計し，開発に要する資源(人・物・金・時間)を見積もる

(3)ユーザの要求定義：システムへの要求を定義し，その機能，性能，運用要件，必要なハードウェア，ソフトウェアを明らかにする

基本制御構造

構造化定理で示すプログラムの基本の制御構造で，次の3つをさす。

(1)順次形(連続構造)(sequence 型)

(2)選択形(条件分岐構造)(IF THEN ELSE 型)

(3)繰返し形(反復構造)(DO WHILE 型)

この他に，次のような2つの追加制御構造がある。

(4)後判定型反復構造(DO UNTIL 型)

(5)多分岐構造(CASE 型)

論理が複雑で，長いプログラムも整理していけば，この3種の構造で記述できる。GOTO文の多用はエラーの原因になりやすく，また開発者以外の人が読みとりにくいという欠点があるため，プログラムを基本制御構造で記述することが望まれている。→(関)構造化定理，選択

基本設計(basic design)

コンピュータシステムの設計において，システムのおおまかな機能と構成を設計すること。予備設計ともいう。現状分析の結果に基づき，その問題点を明らかにしたうえで実現可能なシステムの基本構想を作成する。

基本データ領域(prime data area)

索引順次編成ファイルを直接アクセス装置上に作るとき，その記憶領域を(1)基本データ領域(プライム領域)，(2)あふれ域(オーバフロー領域)，(3)索引域(インデックス領域)に分ける。プライム領域はデータレコードとトラック索引を含む。この領域はすべての索引編成ファイルに存在する。ファイルを作成するときは，レコードをプライム領域に書き出す。後から，レコードの追加や挿入があってこの領域に書き切れないレコードはあふれ域に収容される。→(関)索引域，オーバフロー領域，索引順次編成

基本ファイル(master file)

→(同)マスタファイル

キ

逆アセンブル(deassemble)

記憶領域の機械語命令コードを，それと1対1に対応しているアセンブラ形式の文に翻訳すること。通常はデバッグ時，デバッガのコマンドによって行う。なお，市販のソフトウェアは，著作権保護の観点から逆アセンブルを禁止している。→(対)アセンブル

逆関数(inverse function)

あるアルゴリズム(関数)で暗号化した暗号文を元に戻すための関数。暗号化のための計算は簡単に行え，暗号かぎがない状態では，その逆の復号のための計算は困難で長い時間かかることが暗号解読を防ぐ上で望ましい。このような関数の例に，素因数分解の計算困難性を使ったものがある。この計算には，最高速のスーパコンピュータを使っても何年もかかることが分かっている。

逆順(descending order)

昇順の逆の並び。→(同)降順

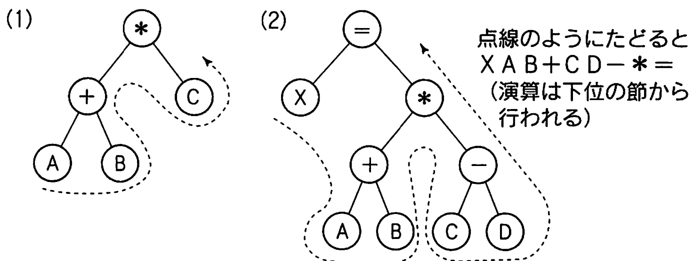
逆ポーランド表記法(reverse Polish notation)

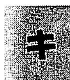
FORTRANやCOBOLなどのソースプログラムをコンパイルする際に行う算術式を機械語に変換する過程では，加減算より乗除算を優先するなどの演算子の優先順位を考慮して，構文を明らかにする。このときに用いる表記法のひとつ。スタックを利用した簡明な算術式の内部表現法を逆ポーランド表記法という。

この表記法では，加減乗除などの演算子をオペランドの後に置く。例えば，AとBを加え，その和にCを乗じることを，AB+C× と書く。また括弧のある場合には，次のように書く。

X=(A+B)×(C−D)→　XAB+CD−×=

この記法は，ポーランドの論理学者 J.Lukasiewiczが使用したもので，括弧なしで数式を表現できる。そのため，算術式の評価が単純に行える。後置表記法ともいう。(1)，(2)は構文木では図のように書けるものである。





キャッシュレスシステム(cashless system)

現金(キャッシュ)に代わり，キャッシュカードやプリペイドカードを使用し，エレクトロニックバンキングシステムにより現金での決済を不要とする社会システム。これが各個人に普及すると，高額の現金を持ち歩かなくても生活できる社会，キャッシュレス社会になる。

銀行のキャッシュカード，信販会社のクレジットカードがその代表例。レストラン，デパート，スーパ，ホテルなどが発行している会員カード，会社や学校内だけで使えるコミュニティカードなどもキャッシュレスの例。決済機能を電子化する電子決済システムが完成し，ホームバンキングが普及すると，本格的なキャッシュレスシステムの時代が到来する。

キャド(CAD：computer aided design)

コンピュータシステムを利用して行う，設計，製図などの作業，またはその支援ソフトウェア。→(同)CAD

キャラクタ(character)

コンピュータシステムが使用する文字，または記号のこと。→(同)文字

キュー(queue)

→(同)待ち行列

行(row／line)

一般に，一連の文字列や数字列が横に並んだもの。

情報処理では次のような例がある。

(1)数学用語では行列の行のこと。列(column)の対語。→(参)行列

(2)ソースプログラムのリストの 1行。→(参)行番号，プログラムリスト

(3)ワープロソフトなどのテキスト表示では，画面上のカーソルの上からの位置，文書のページの上からの位置。

(4)表計算ソフト(スプレッドシート)やリレーショナルデータベースの関係表の場合，データを縦，横にグループ化し，横並びのデータを行とし，縦並びのデータを列という。→(類)タプル，アトリビュート

(5)プログラムの二次元配列の場合，配列の大きさを宣言する先の変数。例えば，A(5，10)は 5行10列の大きさを表す。→(対)列

業界標準(de facto standard)

公式機関による規格ではないが，事実上規格同様に扱える，業界または有力メーカの規格。→(同)デファクトスタンダード

境界要素法(boundary element method：BEM)

ラプラス方程式の近似的な数値解法のひとつ。有限要素法に比べると適用範囲は狭いが，未知数の数を減らして処理時間が短縮できる。

キ

共通機能分割(common function division)

モジュール分割技法の1つで，データの流れに着目する代表的な技法。共通に利用できるモジュールを独立させ，呼び出して使用できるように分割すること。

実際には，STS分割やトランザクション分割によって分割を進めていく過程で行われる。例えば，STS分割やトランザクション分割を進めていくと，いくつかのモジュールに共通の従属機能(エラーメッセージの表示，レコードの印刷など)が必要な場合がある。その共通の従属機能を取り出し，別個のモジュールとして定義することにより，同じプログラム内で同じ機能の重複した記述が避けられる。プログラムがコンパクトになり保守もしやすい。このようなモジュール分割を共通機能分割と呼び，分割されたモジュールを共通モジュールという。なお，機能的強度のモジュールを情報的強度のモジュールに変える場合も，共通機能分割である。

行番号(line number)

ソースプログラムの各行につけた番号。プログラムエディタ固有の行番号はプログラムリストなどに印字するが，コンパイラには影響を与えない。行番号は，かつてのFORTRANやBASICのプログラムでは，行番号が飛越し先や制御の範囲などを示す目印になり制御に関わっていた。ふつう10，20，30というように10きざみにつけ，後からプログラムを挿入，追加時は15や24など，中間の番号をつけることで自動的に行番号順に編集できる便宜を考慮した方式であった。しかし，今日では，FORTRANもBASICも構造化プログラミングが可能な規格に改められ，またスクリーンエディタを使用するため，行番号は制御，編集上の意義を失いつつある。

業務情報システム(business information system)

企業内情報システムのうち，企業活動(業務・職務)の経過(各種の取引，生産と供給など)の情報を記録，蓄積，分析する情報システム。企業経営のなまのデータ処理システムで，企業の計画，行動を決定する基礎データになる。

業務フロー(business flowchart)

新システムの外部設計の段階で作成する，データや処理の流れを明確にするための流れ図。データフローダイアグラムやHIPOなどで表すことが多い。これを製作する段階で，ユーザ要件を十分に把握する。

共有ファイル(common file)

ユーザ部門の複数の人が，長期にわたり共用するファイル。もし事故が生じると多くのユーザに影響する。アクセス管理，バックアップ，保守など，個人ファイル以上にファイル管理を厳格に行う必要がある。

局所変数(local variable)

変数名の有効範囲が，それを定義したブロック内または関数内においてのみ有効な変数。局所変数は大域変数と違い，他のブロックと同一の変数名でも，定義したブロック範囲外では関係がなく，変数名を付ける場合の自由さが増す。局所変数に値を引き渡すためには引数などを利用する。

→(対)大域変数

キ

記録媒体(record medium)

データ，レコード，ファイルなどを記録・保存できる媒体。外部記憶装置に使用するハードディスク，フロッピーディスク，MOディスク，CD-ROM，磁気テープなどの記憶媒体をさす。

記録密度(recording density)

ユーザが実際に記憶媒体に記録したデータ量と本来使用可能な(記録可能な)最大値との比率。→(類)ロードファクタ

記録容量(recording capacity)

記憶媒体に記録できる情報量。フロッピーディスクなどでは，本来の媒体が記録できる最大容量をさす場合と，ユーザやOSが装置の仕様に合わせてフォーマットした後の媒体が記録できる容量をさす場合がある。

均一性(uniformity)

オブジェクト指向言語において，文法的構成要素，データ構造や制御構造が，すべて同様な手法で記述可能であること。例えばSmalltalkでは，カプセル化，継承，多様性などのすべての指定が既存のオブジェクトに対するメッセージとして記述できる。これにより，特に覚えたり，常にマニュアルを参照せねばならないような文法の煩雑さが減少する。

近似アルゴリズム(approximation algorithm)

実用的な範囲で近似解(approximate solution)を早く求めるアルゴリズム。計算機で問題を解くとき，数学的な意味で厳密解(exact solution)や最適解(optimum solution)を求めると，著しく計算量が多くなり実用的でない場合がある。これに代わり，現実的な利用に耐える解を少ない計算量で求めるアルゴリズムが求められる。次のような分野でよく利用される。

(1)連続的な事象(連続関数の方程式など)の数値計算

(2)確率的な事象(ランダムな要素をもつ事象)のシミュレーション計算

(3)不確定な事象(予測値や戦略的な判断のための値)のモデル値の計算

金種計算(counting by denominations)

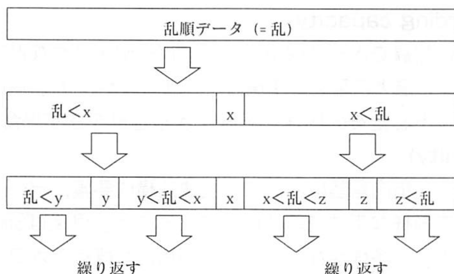
給与等を現金で支給する際に，紙幣や硬貨が何枚ずつ必要かを計算すること。

ク

クイックソート(quick sort)

内部ソートのアルゴリズムの一つ。基本的な考えは，2分探索に似ている。配列中から一つのデータを選び，これより大きなデータを上位に，小さなデータを下位に移動する。データはある値を境に2分割されたわけである。大きいデータ群と小さいデータ群に対して別々に同様の操作を行えば，データは三つの値を境に4分割される。データ群ごとにこれをさらに細かく繰り返せばソートが完了する。分割交換整列法ともいう。

平均計算量は，n件のデータついて $n\log_2 n$ に比例，最悪の計算量はnの2乗に比例するが，実用上よいアルゴリズムと評価されている。なお，このアルゴリズムは改良され，さまざまな高速化アルゴリズムがくふうされている。



クエリー(query)

データベース管理システムで，データベースから検索するデータを指定するルール。システムによって指定法は異なるが，指定のための言語があり，クエリー言語という。→(関)SQL

区分(①segment，②partition)

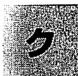
①データやプログラムを分割した単位のこと。→(同)セグメント

②記憶領域を分割した単位のこと。プログラムやデータをディスクに記録するとき，ディレクトリ(メンバ名と格納場所の情報をもつ)を設け，これによって構成メンバ(データやプログラム)を取り出せる方法を用いる(区分編成ファイル)。このとき構成メンバを入れる記憶場所を占有するが，これを区分という。構成メンバが増えると区分も増える。また，削除するとその区分が空スペースとなる。空スペースに入れるのは同じメンバ名のファイルであるが，もとの大きさより大きいものを入れることはできない。

→(参)区分編成

区分化(segmentaion)

1つのプログラムをいくつかの単位に分割すること。主記憶装置の容量の割に大きなプログラムを作る場合に行われる。



区分コード(block code)

コード化の対象をいくつかのブロック(グループ)に分けるとき，ブロック内で連番をつけたコードで，項目を特定の意味のあるブロックに分類したもの(市町村コードなど)。ブロック別コードともいう。

桁数が少ないうえ，空番(欠番＝予備コード)があるためある程度の追加は可能で，グループ集計も可能で分類に便利である。しかし，追加やデータが多くなると不便である。→(参)コード設計

〈例〉大学・高等専門学校コード

0001～0999	国立大学
1001～1999	公立大学
2001～2999	私立大学
3001～3999	国立短期大学
4001～4999	公立短期大学
5001～5999	私立短期大学
6001～6999	国立高等専門学校

区分編成(partitioned organization)

順次編成ファイルをメンバという単位のサブファイルに分割し，別に各メンバの開始位置を示す登録簿を作ったファイル編成法。登録簿のアドレスを見てからデータ域のメンバごとにアクセスする。メンバは順次編成となっているため，分割形順次編成ファイルということがある。ライブラリ形式のファイルは，ほとんどがこの編成法をとっている。

組(tuple)

通常は1件分のデータ。カード型データベースの場合はカード1枚に，リレーショナルデータベースの場合は，二次元表の1行に相当する。

組込み関数(intrinsic function／built-in function)

C言語やFortranにおいて，三角関数，平方根，整数化などプログラム中でよく使う関数をあらかじめコンパイラに組み込んでおき，プログラマはソースプログラムを記述するときに呼び出して引用できる。

クライアント(client)

クライアントサーバモデルのネットワークに接続したコンピュータの中で，他のコンピュータ(サーバ)にネットワークサービスを要求し，サービスを受ける側のコンピュータ，またはプログラム。クライアントは，個人

ク

で使用するワークステーションやパソコンであることが多い。クライアント側のソフトは，ワープロ，表計算，プレゼンテーション，データベースアクセス，ブラウザ，電子メール，グループウェアなど多様なものがある。サーバ側はファイルサービス，データベースサービス，プリントサービスなどを行う。→(対)サーバ

クライアントサーバシステム(client server system：CSS)

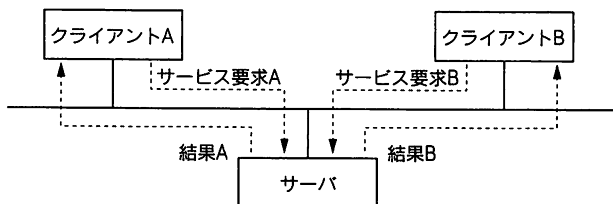
分散処理システムの一形態。処理を要求するパソコンやワークステーションなどのクライアントと，クライアントに各種サービスを提供する専用コンピュータ(サーバ)とをネットワークで接続した情報処理システム。

LAN(構内通信網)などのネットワーク上に複数のクライアントとサーバを置くことも可能で，機能を分担して負荷分散を図ることができる。水平分散処理システムの代表的形態である。クライアントサーバシステムでは，接続しているコンピュータ間に上下関係はなく，相互に受持ちのサービスを提供する役割に変わっている。

クライアントサーバモデル(client-server model)

分散処理システムの一形態。あるサービスを専門に提供するさまざまなコンピュータとそれらのサービスを受けとる複数のコンピュータをネットワークで接続したモデル。

このモデルでは，あるサービス専門のサーバをネットワーク上の別のシステムに複数置くことも可能で，負荷分散を図ることができる。なお，ホストコンピュータ中心の階層的な垂直分散処理と異なることを強調するため，水平分散処理システムの代名詞となっている。

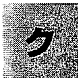


クラス(class)

オブジェクト指向プログラミングにおいて，同一レベルの機能，または性質をもつ異なるオブジェクトの集合。クラスの間に階層関係を定義できる。オブジェクト指向プログラミングは，複雑な大規模システムの開発，保守において本来の効果が現れてくるもので，現実的には，クラスライブラリが充実していることが前提となる。

クラスタ(cluster)

磁気ディスクの物理的な使用単位であるセクタ・レコードに対し，使用・不使用や領域の確保・開放といった管理を効率的に運用するための論理的な最低使用単位。1クラスタに対するセクタ数は機種によって異なる。



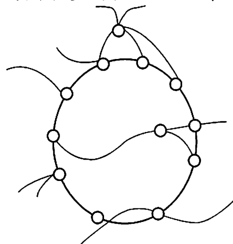
クラック(crack)

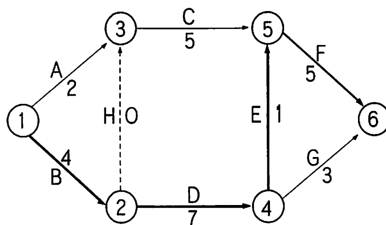
アクセス権のない者がシステム内部に不正に侵入すること。侵入者のことをクラッカーという。類語のハッカーは必ずしも悪い意味で使うとは限らないが，クラッカーは犯罪者に近い意味で使う。インターネットの普及で，全世界のクラッカーからシステムがクラックされる危険がある。クラックの手口は，他人のアカウント(利用者IDやパスワード)を盗み，プログラムの欠陥や運用上の管理不徹底などのセキュリティホールをついて侵入する。その結果，データの盗用，破壊，改竄(かいざん)，捏造(ねつぞう)，ウィルスの混入などを行い，大きな被害を与えることがある。対策としてファイアウォールの設置，アカウントの管理徹底とパスワードの定期的変更が有効である。→(類)ハッカー

グラフ(graph)

データ構造でいうグラフは，距離や長さを無視して，いくつかの点と隣接関係にある点を結ぶ線で表す図。さまざまな事象の間の関係を，事象を点で，事象間のつながりを線で表現したのがグラフである。線に方向をもたせたり，重みをもたせたりすることもできる。例えば，駅を点で表し，所要時間や距離，料金などを線で表す路線地図はグラフである。

点を頂点(vertex)，または節点(node)ともいい，結ぶ線を辺(edge)，孤(arc)，または枝(branch)ともいう。ある点から別の点に結ぶ一連の枝があるとき，この経路を道(path)という。ある頂点から別の頂点を経由して自分自身に帰る道を環，または閉路(cycle／closed path)という。





グラフィカルユーザインタフェース(graphical user interface : GUI)

アイコンやマウス操作を入出力の基本とする視覚的なマンマシンインタフェース。→(同)GUI，(対)CUI

ク

繰返し(repetitive／iteration)

基本的な制御構造のひとつで，同一文(statement)を一定の条件下で何度も反復実行すること。繰返しの回数を制御するため，一定回数の指定や，ループの脱出条件などの設定を行う。反復ともいう。

Repeat Until型とDo While型があり，前者はどんな場合でも必ず1度は処理を実行するが，後者は条件によって，1度も実行しない場合がある。

クリティカルパス(critical path)

PERTに関連する用語。アローダイヤグラム上のすべての経路(path)のうちで，矢線の長さ(所要時間や移動距離，費用など)がもっとも長くなる経路。全体余裕が0のアクティビティを結んでできる経路である。通常，アローダイヤグラム中では太線で表示する。この経路があるために，ほかの工程の進度に関係なしに全体の距離や時間が決まるため，この経路上の作業を重点管理し，クリティカルパスを短くするように生産設備や労力を転用する。

グループウェア(groupware)

ネットワークを使用して，オフィスでの共同作業を総合的かつ円滑に行うためのソフトウェア。おもな機能は，電子メール，電子掲示板，会議のスケジュール管理，住所録，DTP，電子会議システムなどである。従来，電話や文書でやり取りしていた情報交換を端末の前にいながらにして行えるようにすることを目的にしている。具体的には，グループ内の仕事の流れ(ワークフロー)を支援するグループウェア機能を活用する。欧米での研究や製品化が盛んである。なお，グループウェアを使用して共同作業を行うことをCSCW(computer supported cooperative work)という。代表的な製品にロータス社のLotusNotes，ノベル社のGroupWiseなどがある。なお，決済や契約を行うために印鑑捺印処理や認証を行うシステムも登場している。→(参)認証，証明機関

グループトータル(group total)

あるキー項目について整列したデータをそのキーの値ごとに集計する処理。例えば，部課単位や商品単位などのように部門別の集計をするような場合，ひとつのグループとして同じ性質のものを集計し，さらにすべてのグループの合計を求めるような処理のこと。例としては，売上一覧表における商品別売上金額の大計，中計，小計を求めることに相当する。

キー		データ	トータル		
中コード	小コード	データ			
		データ			
		データ			
		データ	小計		
	小コード	データ			
		データ	小計	中計	
中コード	小コード	データ			
		データ			
		データ	小計		
	小コード	データ			
		データ	小計		
	小コード	データ			
		データ	小計	中計	
	小コード	データ			
中コード		データ	小計		
中コード	小コード				
		データ			
		データ	小計		
	小コード	データ			
		データ	小計	中計	大計

グループ分類コード(group classification code)

分類コードの一種で，階層構造のデータを所属するグループごとに分類するコード。コード化対象データを，大分類，中分類，小分類などに分類し，コードの各桁に対応させ，各分類内で一連番号を付ける。桁別分類コード，または単に分類コードともいう。データが多くなると桁数も多くなる。例えば，会社などの組織のコードを部，課，係などに分けて，それぞれに一連番号を付けて用いる場合など。各桁の数字が1グループに対応し特定の意味をもつ。下位の桁を切捨てるか，0に置き換えると上位のコードを得る10進分類コードが代表的。JISには，産業分類コード，職業分類コード，勘定科目コードなどがある。

クローズ(close)

ファイル操作中，またはプログラム中で，ファイルの使用終了時に与える命令。「ファイルを閉じる」ともいう。この命令を受け取ると，ファイル操作終了のための後処理を行う。→(対)オープン

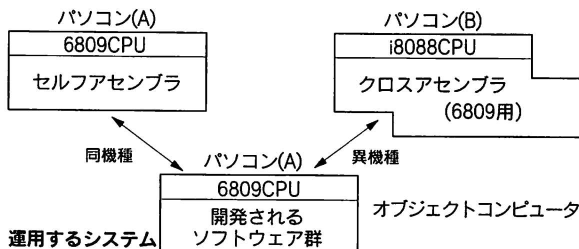
クローズドバッチ処理(closed batch processing)

コンピュータの操作を専任のオペレータに任せるバッチ処理方式で，ユーザは処理したいデータやプログラム，処理手順を一括してオペレータに渡して処理を依頼，処理後のデータやプログラムを受け取る運用の方式。

クロスアセンブル(cross assemble)

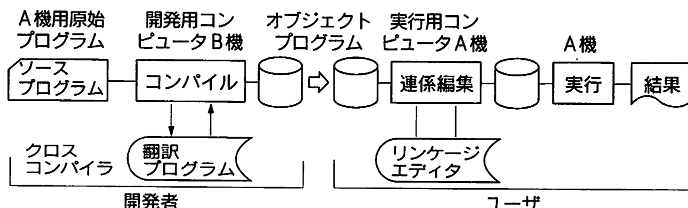
ある計算機用のプログラムを，それとは異なるアーキテクチャをもつ別の計算機上でアセンブルすること。この結果，ある計算機上で働く機械

語を別の計算機上で生成できる。生成したオブジェクトプログラムを実行するコンピュータをオブジェクトコンピュータ，またはターゲットコンピュータという。実行用コンピュータがプログラム開発に自由に使用できないとき，実行用コンピュータを使用するより開発の効率が良い場合などに行う。



クロスコンパイラ(cross compiler)

他の計算機システムで実行するためのオブジェクトモジュールを生成するプログラム。実行させるのとは異なる計算機でコンパイルが可能なため，プログラム開発が効率化できる。特にクロスコンパイラと区別して，自機種用コンパイラをさすときにはセルフコンパイラという。



グロッシュの法則(Grosch's law)

「コンピュータの性能は，価格の2乗に比例する。したがって，大型コンピュータほど価格性能比は良くなり，経済性が高い」という1950〜70年代の経験的法則である。米国IBM社にいたH．グロッシュ氏が唱え，大型コンピュータ普及の推進役を果たした法則である。しかし，この法則はLSIの製造技術の進歩でコンピュータ価格が大幅に低下したことにより，1970年代後半には崩壊した。分散処理時代の現在では，まったく当てはまらない法則になってしまっている。

経営情報システム(management information system：MIS)

企業経営のための情報システムを経営情報システムと呼んでいる。その目的は，省力化や合理化をすすめることにより，生産性の向上をはかり，経営を安定させることである。企業の経営に欠かせない，人事情報システム，生産管理システム，財務会計システム，意志決定支援システムなど，企業内部用の基幹情報システムで構成されている。経営に必要な情報を蓄積する優れたデータベースを構築し，必要なときに必要な情報を入手できる体制が，その基本である。→(参)意思決定支援システム

経営戦略情報システム(management strategic information system：SIS)

企業活動の戦略的な目標や方針，計画などに関する意思決定の際の情報を提供するシステム。従来の経営情報システム(MIS)が経営の効率性に視点を置いて構築してあるのに対し，経営戦略情報システムは戦術的なものを超えて多角化，集団化，合併，海外進出など企業の長期的な戦略に重点を置いて構築する。こうした情報システムが望まれる背景としては情報革命や技術革新の中での企業の国際化，複合化，巨大化がある。

→(参)経営情報システム，意思決定，意思決定支援システム

継承(inheritance)

広義には，階層構造の構成要素の上位の概念がもつ性質を下位の概念にも適用すること。オブジェクト指向プログラミングにおいては，あるクラスの精細化(メソッドやインスタンスの指定)によって別のクラスを定義できるという特性。この新たなクラスは，さきのクラスの属性(機能や性質)をそのまま継承する。新しいクラスの定義は，目的とする処理とさきのクラスで定義された処理との差を記述するだけで済むため，充実したクラスライブラリを用いれば，新しいプログラム作成に必要な記述量は少なくて済む。

経歴ファイル(history file／historical file)

システムの稼働中，どのようなプログラムがどのファイルを対象に稼働し，処理がどのように終了したかなどを記録しておくファイル。履歴ファイル，ジャーナル，ログともいう。通常磁気ディスク中に確保する。

経路(path)

→(同)パス

ケーブルテレビ(cable television：CATV)

有線テレビ。アンテナを使わず，中継局とケーブルで接続し，放送を受信する。初めは難視聴地域の解消を目的に導入した。次第に提供情報の多

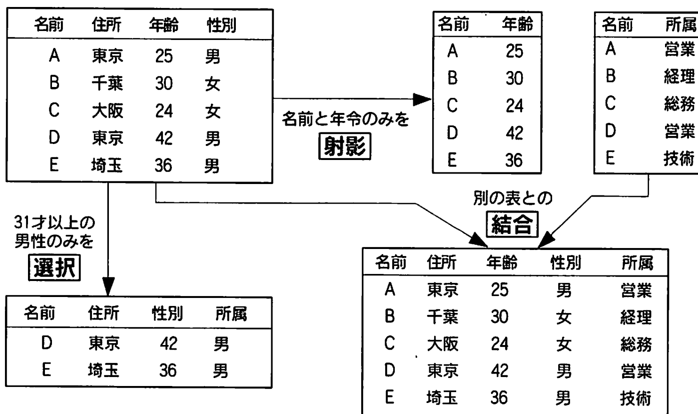
様化や双方向通信などの機能を追加し，一般でも視聴料を払って使用するようになった。有線であるためチャンネル数を多くできるメリットがある。ユーザがケーブルを設置するには業者との契約が必要である。業者は，有線電気通信事業法による届出が必要であり，有線テレビの放送を行うには有線テレビジョン放送法に基づきCATV運営者としての許可を受ける必要がある。CATV用のケーブルは10メガbps以上の高速通信が可能であり，番組提供以外の通信網としても注目されている。

ゲームの理論(theory of games)

企業間の競争など，互いに競合する利害関係をもった複数の者の間で相互に競い合う条件のもとで，利益を最大にする解をさがす理論。オペレーションズリサーチの分野では，相手がどのような手段を選んだときも，自分自身が最も有利になるような手を発見する理論をさす。

結合(join)

関係データベースのデータ操作の一機能。元になる表と比較する表とで共通する属性を調べ，設定した条件を満たすものを取り出し新しい表を作成する。射影と選択を一度に行うような機能にあたる。→(参)関係代数



名前	住所	年齢	性別
A	東京	25	男
B	千葉	30	女
C	大阪	24	女
D	東京	42	男
E	埼玉	36	男

名前	年齢
A	25
B	30
C	24
D	42
E	36

名前	所属
A	営業
B	経理
C	総務
D	営業
E	技術

名前	住所	性別	所属
D	東京	42	男
E	埼玉	36	男

名前	住所	年齢	性別	所属
A	東京	25	男	営業
B	千葉	30	女	経理
C	大阪	24	女	総務
D	東京	42	男	営業
E	埼玉	36	男	技術

結合子(connector)

流れ図記号の一種。フローチャートの流れ線の切れ目と，その続く場所を表す。フローチャートを作成する際，流れ線が交差して見にくくなったり，一枚の用紙に入りきらない場合に，流れ線の中断する所に出口の印をつけ，流れ線が再会する所に入口の印をつける。出口と入口は，結合子の中に数字や文字を記入して対応づける。→(参)流れ図，流れ図記号

結合テスト(combined test／joint test)

関連する2つ以上のプログラムモジュールを連結(結合)して，正しく動作するかどうかをテストすること。個々のプログラムが仕様書の機能どおりに動作するか調べるモジュールテスト(単体テスト)で正常に動作しても，結合してみてはじめて，エラーを発見するケースがあるので，モジュールの結合順序や結合時期を明確にしながら行う必要がある。

→(参)モジュール，モジュールテスト

決定表(decision table)

→(同)デシジョンテーブル

原因分析(analysis of factor)

目標と現状の間にある障害の要因，条件，現象などを分析する活動の総称。問題の分野によりさまざまな手法が開発されているが，問題の定義と目標の設定で分析の着眼点が異なるのが普通である。一般的には，統計的分析，経済性分析，安全性分析，信頼性分析，シミュレーション解析，最適化モデルとの対比という手法がある。仕事の流れについては，工程分析，作業分析，動作分析，時間分析などの手法がある。

限界検査(marginal test)

初期障害をもつ構成要素を検出し指摘するために，供給する電圧や周波数などの動作条件を公称値付近で変化させる技法。ハードウェアの安定性・余裕度を確めることができる。

言語プロセッサ(language processor)

プログラム言語で記述したプログラムの翻訳や解釈，変換などを行うプログラム。言語処理プログラム，翻訳プログラムともいう。ある言語を別の言語に，あるプログラム言語を別のプログラム言語に翻訳する。

→(例)コンパイラ，インタプリタ，ジェネレータ，アセンブラ

検査(test)

→(同)テスト

検索(①retrieve，②search／find)

①データベースの中から必要な情報を体系的に整理・分類しながら引き出すこと。検索キーに種々の条件指定をつけることで，特定のデータを迅速，かつ正確に再編成，あるいは表示することができる。例えば，商用データベースの情報検索システムでは，キーワードの記述，条件の指定などを対話的に行える専用のコマンド言語がある。→(類)情報検索，類義語検索

②コンピュータシステムのファイル管理機能，データベースソフト，ワー



プロソフトの文字列探索機能のひとつ。対象とする装置，ディレクトリ(フォルダ)，ファイルから指定した文字列を探し表示すること。ワイルドカードや正規表現を使ったり，条件式を使ったりして検索できるソフトが増えている。→(関)あいまい検索

検査プログラム(check program)

コンピュータが正しく働くかどうか，またソフトウェアが間違いなく作られているかどうかを確かめるためのプログラム。例えば，トレーサはプログラムの指定した命令を実行するつど，命令コード，オペランド，オペランドで参照している番地の内容などを印刷する。またダンプルーチンは，主記憶装置の全内容や指定した特定領域の内容，さらに使用中のレジスタの内容，標識やスイッチの状態を印刷する。→(参)ダンプ，トレーサ

原始プログラム(source program)

→(同)ソースプログラム

検収(acceptance)

システムの開発・テストが終了し，開発部門から運用部門に移行できると判断すること，またはシステムを受け入れること。

検証(verification)

プログラムの正当性(correctness)を証明すること。プログラムの正当性とは，アルゴリズム，またはそれを表現したプログラムが，与えられた仕様を満たしていることである。→(関)正当性

現状分析(system analysis／status quo analysis)

システム開発に先立ち，現行のシステムで情報がどのように処理されているかを知ることを目的として行う調査。これによって，現行のシステムにどのような問題点があるかを具体的に知ることができ，またいろいろな業務活動についても知ることができる。→(関)事務分析

件数検査(count check)

入力データの重複やもれをチェックするために，原始データの件数を調べる方法。伝票の枚数などを人間が数えて入力し，いっぽう，システムも入力データの件数を数えて，一致するかどうかを確かめる。

源泉／変換／吸収分割

(Source／Transform／Sink division：STS division)

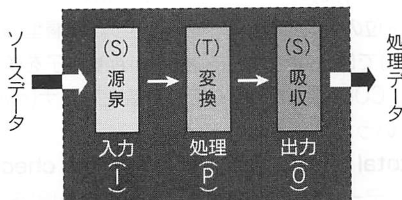
モジュール分割技法は，データの流れに着目するものとデータ構造に着目するものとに大別されるが，前者に属する代表的なものの一つがこの技法である。G．マイヤーズが提唱した。複合設計ともいうが，一般には，英字の頭文字からSTS分割と呼ぶ。プログラムを，ほぼIPOに対応した次

の3つに分割し，それぞれを一つの機能として記述し，モジュールとして定義する。したがって，データの内容により，処理が多岐に分かれるようなものには不適である。→(参)最大抽象点

(1)源泉部分(Source：入力データを処理する部分)

(2)変換部分(Transform：入力データを出力データに変換する部分)

(3)吸収部分(Sink：出力データを処理する部分)



原票設計(source document design)

データは，原始帳票(原票)から，画面を見ながらキーボードでディスクなどの入力媒体に変換することが多い。その基本となる原始帳票のレイアウトは，通常次のような点に注意して設計する。

(1)必要十分なデータを記入順序を考え，重複をさけること

(2)関連帳票との記入位置，記入項目を統一すること

(3)ファイリングしやすい形にすること

(4)キー入力などの作業がしやすい並びにすること

減分(decrement)

ある値から一定の値を繰り返し減算していく場合の，減算する単位値のこと。例えば，プログラムで，ループを1回実行するたび制御変数やポインタから一定の値を減算する場合，減算していく値が減分である。なお，同様に加算する場合には増分という。減分は負の増分ともいえる。

→(対)増分

子(child／member)

①木構造において上位の節を親と呼ぶときの，下位の節。特に順序木(ordered tree)は，一つの節が子をいくつももち，その子の間に順序関係がある。

②親子集合型(set type)のデータベースで，相対的に上位のレコードを親レコード，下位のものを子レコードという。階層型の代表的なDBMSであるIMS(IBM)では，親をペアレント(parent)，子をチャイルド(child)という。また，CODASYLのDBTGでは，親をオーナ(owner)，子をメンバ(member)という。

合計検査(batch total check／total check／sum check)

入力に際し，データとその合計(手計算より得た値)を入力し，計算機側でも入力した値により，別途合計を計算し，照合することで，入力の誤りやデータのもれ，過不足などを検出する検査法。→(参)ハッシュ合計

降順(descending order)

データのソート処理で，配列やテーブルなどの特定のキー項目について大きい順から小さい順へ並ぶ順序。昇順の反対の順序。文字列の場合，ZからAへの順序になる。逆順ともいう。→(対)昇順，(参)ソート

更新(update)

①すでにあるファイル中のレコードの追加や削除，変更などの処理。アップデートともいう。内容を変更した新しいファイルで書き換えること。同一ファイル名で保存すると古いほうのファイルは読み書きができなくなる。→(同)ファイルの更新

②ソースプログラムをデバッグ，または機能向上などを目的にして修正すること。新しいソフトウェアをインストールし直すことを含む。

高信頼化技術(technology for highly reliable systems)

コンピュータシステムの信頼性を高めるためのハードウェア，ソフトウェア両面の技術的施策の総称。

(1)ハードウェア面：システムの故障率の低減，予防保全，多重化など

(2)ソフトウェア面：より厳しいテスト技術，バックアップ技術，セキュリティ技術→(参)信頼性評価尺度，信頼性

更新履歴(update history)

更新は，ファイル中のレコードの追加や削除，レコード中の項目の変更などを行うこと。アップデイトともいう。履歴は，システム稼働中，どの

ような処理をどの使用者が行い，処理がどのように終了したかなどを記録しておくこと。記録するファイルを履歴ファイル(historical file)という。更新履歴の保存は，誤った処理，ファイルの追加や脱落などの事故にそなえる措置である。プログラムやデータなどのマスターファイルを更新する場合，更新後も前回，前々回のファイルを定めた世代(または定めた期間)にわたって保存するのが通例であり，こうしたファイルの管理法を世代交替法(generation system)という。

高水準言語(high level language)

汎用プログラム言語を大きく「機械向き言語」と「問題向き言語」に分けたときの問題向き言語。機械向き言語と違い，プログラミングの容易な言語で，主としてC，C++，Pascal，COBOL，FORTRANといったコンパイラ言語をさす。機械に密接した言語ではなく，ハードウェアからの独立性の高い，より人間の思考に近い言語と言える。逆に機械向き言語は機械語やアセンブラ言語のことをさし，機械に密接した言語で，低水準言語という。→(参)問題向き言語，(対)機械向き言語

合成コード(synthetic code)

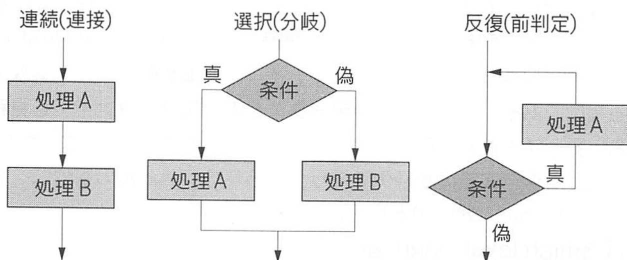
コード化の一技法で2つ以上の異なるコード体系を組み合わせてコードとしたもの。表意コードとの組合せが一般に多い。例えば，学籍番号をあらわす場合，学部(表意コード)＋入学年度(表意コード)＋クラス(表意コード)＋出席番号(順番コード)などの複数のコード体系を用いる。多目的利用は可能であるが，桁数が多くなりがちである。

構造化設計(structured design)

大規模なソフトウェアを誤りなく効率的に開発するプログラミング規範として，また，既存のプログラムの保守に多くの時間を費す事態に対処するため，プログラムの内容を把握しやすくするために考案したプログラムの設計方法。大規模なプログラム開発時には，プログラムの処理機能を分割し，モジュール化する。各モジュールをさらに機能単位に分割するなどして階層構造をもたせる。このようなモジュール構成でプログラミングする手法を構造化プログラミングという。→(関)モジュール，階層構造，段階的詳細化，トップダウンプログラミング

構造化定理(structure theorem)

計算機科学者C.BohmとG.Jacopiniが1966年に発表したプログラムの構造原理。入り口ひとつ，出口ひとつのプログラムは3つの制御構造の組合せですべて表現できることを示す。構造化定理に基づいて作成したプログラムを構造化プログラムという。



構造化定理に従って作成されたプログラムは，個人差が少なくなるために読みやすく，構造が見やすく整理できているために，検証しやすくなり，結果的に信頼性の高いソフトを作ることができる。

→(関)NSチャート，PAD，選択

構造化プログラミング(structured programming：SP)

①大規模なソフトウェア開発においては，構造化設計と同義。

②小規模なプログラム開発，またはモジュールやサブプログラムの開発においても，プログラムの機能を分割し，制御構造を明らかにするのが望ましい。手続き型言語を使用するなら，(1)順次形構造，(2)選択形構造，(3)繰返し形構造の 3つの基本制御構造を用いれば，GOTO文を使わずにプログラムが作成できるという数学的裏付け(構造化定理)がある。

構造体(structure)

データ型の異なる複数の値で構成したデータ構造をレコード型または構造体という。1つの名称で参照できる複数の変数の集合体の形態のひとつ。個々の構造体の要素(メンバ)はメモリ上の連続した領域に確保される。

工程管理(production control／process control)

①一般には，工業製品を計画的に生産するために，生産資源を総合的，経済的に管理する手法。建築，土木など単品を完成させる工事では，資材，日程の管理。品質管理では，品質を左右する要因を制御し，不良品を作らないようにする活動。

②情報処理では，システム開発工程全般を通じての人材管理，資源管理，文書管理，日程管理，品質管理などの管理項目について最適化を図ること。

工程設計(process planning)

工業製品を製造する工程のうち，設計工程と製造工程の間に行う作業。設計データを分析し，加工法や加工機械を決定し，製作作業に入れるようにする。設計技術と加工技術の双方に通じている技術者が行う。

構文(syntax)

プログラムの形式的な構造に関する文法規則。特にコンパイラ言語などの高水準言語では，ソースプログラムを正しく構成するための文法を厳密に規定してあり，誤った文は翻訳することができない。そこで，コンパイラの中には必ず字句や構文チェック(syntax check)の機能がある。コンパイラは，コンパイル中に翻訳不可能な誤りを検出すると，誤りの場所，種類などを出力する。これをコンパイルエラーメッセージという。

構文解析(syntax analysis／parsing)

①機械翻訳や自然言語処理で，入力した文がどのような文法構造になっているかを分析すること。解析法としては下向き法や上向き法などがあり，前者の例はLL(k)法が，後者の例はLR(k)法などである。構文解析プログラムのことをparserという。

②ソースプログラムのコンパイル時，コンパイラは，ソースプログラムを分析する。英数字などの識別子，定数，演算子などの構文を構成している要素に分解した後(字句解析)，それらがどのように組み合わさっているかを調べ(構文解析)，オブジェクトプログラムを生成する。コンパイラは通常構文解析と同時に，あるいはその後，意味解析も行う。

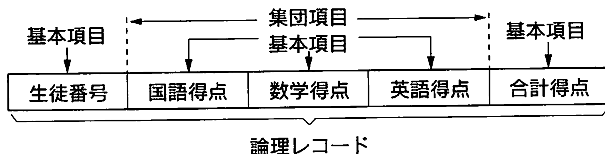
→(参)意味解析，字句解析

構文木(syntactic tree)

ソースプログラムの文や句の構造を木構造で表現したもので，木の各節点には文法的なカテゴリ名(例えば，演算子名)が与えられ，葉には単語名(例えば，変数名や定数名)が与えられ，根にはその字句列全体のカテゴリ名(例えば，代入命令)が割り付けられる。根を上に，葉を下に書くのが通例。代表的な構文木に算術式の構造を表す逆ポーランド表記法がある。

項目(item)

データ集合の構成要素のこと。例えば，ファイルはレコードのようないくつかの項目からなり，さらにそのレコードは他の項目からなる。それ以上細分化できない項目は基本項目といい，基本項目が集まってできた要素は集団項目という。



→(類)変数，フィールド，(関)基本項目，集団項目

コ

コ

効率(time efficiency)

アルゴリズムの評価基準の一つで，答えを得るまでの手間(complexity＝計算に要する実行命令数)が少ないこと。または，そのアルゴリズムの評価関数。手間や時間計算量で評価する。特にソートや探索の処理は，アルゴリズムやデータ量によって，効率が著しく異なるため効率の観点から評価することが大切である。→(参)手間，時間計算量

コーディング(coding)

①流れ図などで表した手順にしたがって，プログラム言語を用いてプログラムを記述すること。言語がコンパイラ言語であってもアセンブラ言語であってもコーディングという。→(類)プログラミング

②EDPの対象となるデータの項目にコード化をすること。ファイル設計の重要な要素となる。→(同)コード化，コード設計

コーディングシート(coding sheet)

コーディングに用いる用紙で，プログラム言語の書式に合わせた固有の形式を印刷した用紙。プログラム開発に必要な諸情報を記入する欄を設けてあるのが普通である。コーディング用紙ともいう。

コード(code)

①(情報交換用符号)文字データの表現方法を規定するあいまいさのない規則や体系。例えば，JISコード，ASCII，EBCDICコードがある。

②(データコード)データを識別するために，使用目的に応じて，識別，分類，あるいは配列するために，事象名につけた記号や略号。例えば，データベース利用の際，キーワードに代えて利用できる文字列。キーワードとコードの間には対応があるが，複数の分野で使用する同一のキーワードなどには複数のコードを与えることがある。→(参)コード化，コード設計

③(プログラムのコード)データの所在や命令の種類をある形式に変換したもの。→(参)命令コード

コード化(code／encode)

データ項目(例えば，製品の名前や取引先の名前など)に一定の方法で，記号または符号の列を対応させ，簡潔な別の表現にしてデータ処理を行う。この対応づけをコード化，またはコード設計という。→(類)コード設計

コード設計(code design)

データ処理に際して，データを識別，分類，区分，照合，あるいは配列を容易化，効率化するために，データに一定の基準を設け体系化を図ること。コード設計でいうコードとは，データコードのことで，使用目的やデ

ータ量に応じて，事象名につけた符号である。通常，次の順で行う。

(1)処理すべきデータの選定(コード化の対象など)

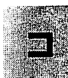
(2)データの体系化(識別・分類・配列)

(3)コード化の調整と決定(表示のわかりやすさ・エラー検出など)

(4)コードの文書化

(5)コードの管理

コード化の方式には，順番コード，区分コード，分類コードなどがある。



コールドスタート(cold start／cold boot)

計算機システムを，そのシステムがもつ初期値を使い，初めから起動すること。その時点でコンピュータの主記憶装置が保持していた情報は失われる。システムが暴走した後，最初から処理を開始する必要があるときなどに行われる。→(対)ウォームスタート

コールドスタンバイ方式(hot stand-by system)

2組のシステムを用意し，一方のシステムの障害発生時に，別のシステムで対応するシステム構成。信頼性が厳しく要求されるオンラインシステムでは，デュプレックス構成にしてシステムの障害時にも稼働し続けるようにする。しかし，即時に切替えるデュアル構成システム(ホットスタンバイ方式)に比べ，主系ダウン時の切り替えに時間がかかることから，コールドスタンバイ方式という。→(対)ホットスタンバイ

互換(compatible)

部品，機器，装置等が同一の機能を有し，差し替え，置き換えが可能なこと。コンピュータや周辺装置などでは，OSやプログラム，JCL，データ形式など同一のものが使える機種を意味する。ハードウェアでは，内部の構造が異なってもICなどの素子がそのまま差し替えて使用できる場合をピンコンパチブル，装置全体が互換でケーブルや電源を付け換えるだけですむ場合はプラグコンパチブルという。

互換機(compatible machine)

ある機械と同等の性能や機能をもつ機械。一般には，オリジナルの機械で動作するソフトウェアや周辺機器をそのまま使用することができる。しかし，まったく同一の仕組みではないため，一部の機能がそのままでは動作しないこともある。

互換性(compatibility)

あるハードウェア，もしくはソフトウェアを別のものと置き換えたとき，オリジナルのものと同等に機能すること，または，その度合を示す。

コ

顧客管理システム(customer management system)

顧客管理は，商品の顧客のニーズを把握し，ニーズを満たす企業戦略を決定すること。狭義には，企業と取引関係にある顧客(株主，資材の仕入先，配送業者，製品の消費者などを含む)の情報を管理すること。顧客の情報としては，顧客属性情報，取引情報(販売実績，引合いなど)，付帯情報(信用，家族，案内発送など)がある。顧客管理システムは，これらの諸活動を実施，または支援するコンピュータシステム。

個人ファイル(personal file)

ユーザ部門の個人が所持，管理するファイル。重要なファイルではあるが，管理が行き届かないことも多いので，データ破壊，盗難，紛失などのセキュリティ管理に気を使う必要がある。

固定長レコード(fixed length record)

ファイル中の論理レコードが，すべて同一の長さに記録してある形式。一般的な形式で，プログラム上も扱いやすい。項目長さが異なる場合でも空白文字を埋め込んで規定の長さにするため，記憶領域の利用効率は必ずしもよくない。→(対)可変長レコード，(参)相対編成

コネクタ(connector)

流れ図記号のひとつ。処理の入口，出口，中断した流れ線の接続を明らかにする記号。→(同)結合子

コピー(copy)

原典(オリジナル)のデータはそのまま残して原典からデータを読み取り，ほかの場所へ書き込むこと。書き込んだデータの物理的な形は原典での形と異なっていてもよい。事故や記憶媒体の劣化に備えるために行う。→(参)媒体変換，バックアップ，ミラーリング

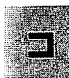
コピープロテクト(copy protect／copy protection)

ソフトウェアの違法なコピーができないように供給者側が原典に加工すること。コピー用のプログラムがエラーを起こすような情報を付加したり，原典のプログラムの使用機を限定するような付加機能を付けるなどの方法をとる。

コマンド(command)

広義には，命令，指令のこと。またその信号を含む。ユーザが直接コンピュータシステムやアプリケーションに与える命令。主としてキーボードから入力したり，バッチプログラムに記述する命令をさす。主記憶に常駐するコマンドプログラムに組み込んでいるコマンドを内部コマンド，外部記憶装置に置いてあるコマンドプログラムで使用できるコマンドを外部コ

マンドという。また，アイコン操作などにより，コマンドを意識しなくてもパソコンを操作できるようになった。これらの操作も対応するコマンドとオプションパラメタに置換して実行しており，コマンドの一種といえる。



コマンド方式(command mode)

商用データベースシステムの利用形態。利用者が通信回線を介してDBMSに働きかけ，対話的にデータベースのデータを操作する方式。例えば，情報検索システムではキーワードの記述，条件の指定法など専用のコマンド言語がある。

コメント(comment)

文書やプログラム中に記述する補助的なメモ書き。ソースプログラムの場合，コンパイルの対象とならない。文書の場合は印字されない。
→(同)注釈

コラボレーション(collaboration system：CS)

もとは科学技術ネットワークの用語で，未解決の大きな問題の研究のために多数の研究者が協同して利用するデータベースやコンピュータ資源をさす。転じて，同様の協調的なグループの作業を支援するシステムをさす。　共有データベースには参加者が自分の実験結果や考察，意見などを書き込みメンバに公開し，討論の場ともなる。→(参)CSCW

コンカレント処理(concurrent processing)

並行処理。ある時間間隔内に複数の動作を同時に行う処理形態。CPUが，処理の遅いプリンタなどの入出力装置の動作終了を待たずに次の命令を実行することにより，CPUの遊休時間を減らし，システムを効率よく利用する。マルチプロセッサシステムのように複数のCPUで同時に複数の命令を実行する処理形態(並列処理)も含めてコンカレント処理という場合もある。→(類)並行処理

コンテンツ(contents)

本来は，通信ネットワーク上で交換する情報から宛先情報や制御情報を除いた「中身の情報」の意味であったが，近年は，ソフトウェア(プログラム)に対比して，インターネットで伝達する情報やCD-ROMに納める情報そのものをさす語として用いる。

コントロールブレイク(control break)

キー項目について整列されているデータを処理する際に，その項目の値が変わった時点で，小計の印字など通常の処理と違う処理を行わせる手法。

コンバージョン(conversion)

①あるシステム(A機)のソースプログラムを別のシステム(B機)でも利用し

たいとき，A機のソースプログラムをB機に適合するソースプログラムに変換すること(プログラム変換：program conversion)。
→(同)プログラム変換，(参)移植性
②入力したデータと処理可能なデータの形式やコードが異なる場合，処理のために入力データを処理可能なデータ形式へ変換すること。特にある符号体系から別の符号体系に変換すること(コード変換：code conversion)。

コンバータ(converter)

データやプログラムを加工して，別の形式に変換するハードウェアやソフトウェア。フロッピーディスクで，8インチディスクから3.5インチディスクへのコンバート，2Dの記憶密度のディスクから2HDの記憶密度のディスクへのコンバードをする場合はハードウェアによる物理的コンバータを使う。同じディスクでも使用するオペレーティングシステムによってはデータの記録の方法が異なるため，相互に利用するために論理的コンバートをする必要がある。同じオペレーティングシステムでもデータの形式が異なる場合はコンバートしなければ使用できない。

コンパイラ(compiler)

FORTRAN，COBOL，Pascal，Cなどの高水準言語で書いてあるソースプログラムを機械語に翻訳するプログラム。コンパイラは，それぞれの機械に固有のものを開発し，メーカ，あるいは言語開発者がユーザに提供するのが普通である。→(類)プリプロセッサ

その種類や機能は多種多様で，さまざまな分類ができる。

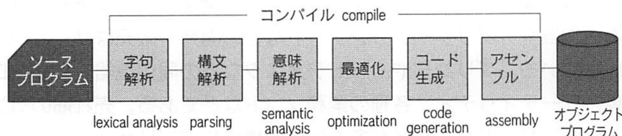
(1)一括コンパイラ：実行に必要なサブルーチンや副プログラムをまとめて，一度にコンパイルする。例)Pascal，C，C++
(2)分割コンパイラ：大きなプログラムをモジュール単位に分けてコンパイルし，後からリンクする。例)FORTRAN，COBOL
(3)クロスコンパイラ：実行用でないコンピュータ上で，実行用コンピュータの機械語コードを生成する。
(4)セルフコンパイラ：実行する自機種で機械語コードを生成する。
(5)マルチパスコンパイラ：コンパイルの過程で，中間的なコードを何回か生成し，最後にオブジェクトプログラムを生成する。
(6)ワンパスコンパイラ：1回のパスで，ソースプログラムからオブジェクトプログラムに変換する。
(7)プリコンパイラ(プリプロセッサ)：拡張した言語仕様で記述したソースプログラムを標準仕様のソースプログラムに変換する。
(8)最適化コンパイラ：コンパイル時間はかかっても，実行時間の短縮を

重視する機械語コードを生成する。

(9)チェッキングコンパイラ：エラー検出，自動修正の機能を強化したコンパイラを特にという。

コンパイル(compile)

プログラム言語で記述したソースプログラムを，コンピュータが実行できる機械語コード(オブジェクトプログラム)に変換する処理。翻訳ともいう。この過程は複雑で，数段階に分けて中間的なコードを作って行うのが一般的である。



コンピュータシステム(computer system)

狭義には中央処理装置，主記憶装置，制御装置，入力装置，出力装置，通信装置などをもつ処理装置の集まり。広義にはソフトウェア，さらにコンピュータ利用の目的で定めた規則や組織化した人材の集団まで含む。

コンピュータネットワーク(computer network)

通信ネットワークを介して相互に連結した複数のコンピュータシステムで構成する複合体。単にネットワーク，または網とも呼び，通信回線の途中に配置した中継機や交換機を含む。

コンポーネント(component)

構成要素のこと。VSAM編成ファイルでは，レコードを記憶するための連続した複数のCA(制御域)をコンポーネントと呼ぶ。ESDS，RRDSでは，クラスタはデータコンポーネントだけから構成され，KSDSのクラスタは，データコンポーネントとインデックス(索引)コンポーネントから構成される。

サーチ(search)

→(同)探索，(類)テーブルサーチ

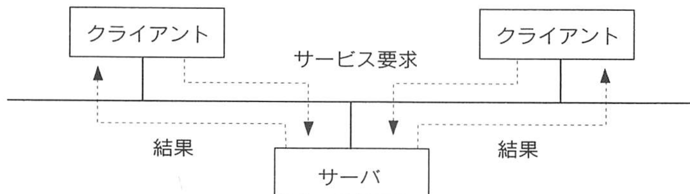
サードパーティ(third party)

コンピュータ本体のメーカや直系の販売会社以外で，そのコンピュータ用の周辺機器やアプリケーションソフトを開発，販売している企業の呼称。サードはメーカでもユーザでもない第三者の意。一般にハードウェアメーカは本体と一部の周辺機器，OSを自社開発するだけである。ほとんどの周辺機器やアプリケーションソフトなどの開発，販売はサードパーティに任せている。プリンタやハードディスクなどの周辺機器，ワープロや表計算，ユーティリティ，各種の業務ソフトなどである。売れ筋のコンピュータほど，サードパーティが多くの周辺機器やアプリケーションソフトを販売し，その機種のシェアの増大につながっている。→(参)ベンダ

サーバ(server)

①クライアントサーバモデルのネットワーク上で資源を保持し，クライアントが出すさまざまな要求を受付け，サービスを提供するプログラム，またはコンピュータ。サーバマシン(server machine)ともいう。

サーバが保持する資源には，データファイル，プログラムファイル，プリンタ，モデム，FAXボード，データベースなどがある。サーバの種類には，ネットワーク上のクライアントに対して，ファイルを共有するファイルサーバ，クライアントの印刷指示により印刷を行うプリントサーバ，電話回線やFAX網，ホストコンピュータとのデータを交換する通信サーバ，クライアントの指示によりデータベースを検索，交信するデータベースサーバ，その他のアプリケーションサーバがある。また，インターネットの普及でWebサーバ，メールサーバ，セキュリティサーバなど多様なサーバも一般化してきた。通常はこれらの機能を1台のコンピュータで行うが，分散させて使うこともある。→(対)クライアント



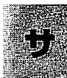
②待ち行列理論においては，サービスを提供する窓口。→(関)窓口数

サーバマシン(server machine)

クライアントサーバモデルのネットワークに接続したコンピュータの中で，各種サービスをクライアントに提供するホストコンピュータ。

サービスアビリティ(serviceability)

→(同)保守性



再エンジニアリング(reengineering)

→(同)リエンジニアリング

再帰(recursive／recursion)

問題を分割統治の手法で分割したとき，分割した部分に，同じ分割統治の手法を繰返し適用すること。アルゴリズム(関数，または手続き)が再帰的であるとは，その内容を定義する際に，自分自身をその定義中で使用することをいう。例えばサブルーチンの実行中に自分自身をサブルーチンとして呼び出すこと。

在庫管理(inventory control)

在庫とは，生産活動中の原料，材料，部品などの不足，枯渇を招かないためのストックである。在庫管理は，在庫管理費用を必要最小限にして，しかも生産活動を安定させるために在庫量を適切に管理すること。

在庫管理システム(inventory control system)

在庫管理を行う制度，ルール，コンピュータシステム。具体的には，在庫モデル(inventory model＝在庫管理システムを研究，分析するときに用いられるモデル)に基づいて，需要と供給の構造，発注から調達までの期間，在庫保管場所の能力，保管費，発注費，品切れの度合と損失，在庫回転率，流通経路などの要因を管理し，発注間隔，有効在庫，発注点，発注点方式を決定するコンピュータシステム。

最上位モジュール(main module)

階層構造の最上位のモジュール。具体的には，プログラムの起動により最初に呼び出されるモジュール，プログラムの入口となるモジュールである。例えば，カウンタの初期化，ファイルのオープンとクローズ，各機能(モジュール)の制御などの処理は，一般に最上位モジュールで行う。これに対し，呼び出されるモジュールを下位モジュール，従属モジュールという。→(参)メインプログラム，(対)従属モジュール，(参)セグメント

再使用可能プログラム(reusable program)

一度主記憶上にロードされた後は繰り返し実行できるルーチンまたはモジュール。その実行中に自らの内容を変更しないように，あるいは実行中に変更した場合は初期状態にもどるように作ってあり，再ロードしないで

も使用できる。実行中に自分自身を修正，変更して終了するプログラムは，再使用できず，再び原典からロードしなおさなければならない。これらは再使用不可能(不能)プログラム(non-reusable program)という。

あるタスクがそのルーチンまたはモジュールを使い終わって初期化がすめば，別のタスクが再使用できるプログラムを逐次再使用可能プログラムという。逐次再使用可能プログラムでは初期状態にもどるまでは，他のタスクに使用させないように排他制御を行う必要がある。また，実行中に自らの内容を変更せず，ひとつの制御の流れ(通常タスクという)が抜け出ないうちに次の制御の流れが入ることを許すプログラムを，特に再入可能プログラムという。→(参)再入可能プログラム

最小木問題(minimum spanning tree problem)

グラフの頂点を結びうる道を保持したまま，道を消去し，最後に残った道の重みの総和を最小にする問題のこと。通信網建設の問題に例えれば，通信可能な経路を維持しながら，省略できる通信線を建設しないことにする場合，最小の費用で全局を結ぶ通信線の場所を決める問題に相当する。この問題を解く代表的な解法にプリム(Prim)のアルゴリズムがある。出発点から枝を伸ばしながら重みを加算して行き，最小の経路を選択する。

最小自乗法(method of the least squares)

たくさんの測定データの組に曲線の当てはめを行うときに用いられる手法。曲線を当てはめるときは，この曲線が各点に最も近いようにすることが必要である。そのためには，各測定点と曲線との差の二乗の合計が最小となるような係数を求める。実際には正規方程式(normal equation)という連立方程式を解く。

最大抽象点(maximum abstract point)

STS分割を行う際の境界点。源泉(S)から変換(T)への境界は，入力データが形を変え，もはや必要でなくなる点であり，変換(T)から吸収(S)への境界は，データの処理が終わり，出力データの形になった点である。モジュールは最大抽象点を境に分割される。→(関)源泉／変換／吸収分割

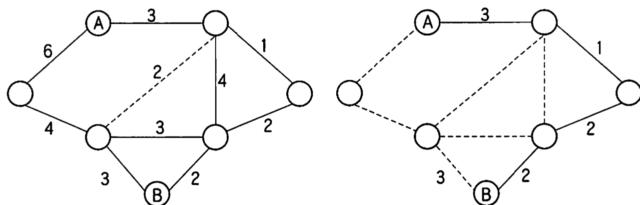
在宅勤務(residence office／telecommute)

通信設備やOA機器を備えることで，出社して業務を遂行するのと同様な環境を作り，自宅で勤務する形態。→(関)サテライトオフィス，SOHO

最短路問題(shortest path problem)

グラフの2点間を結ぶ道の中から，重みの総和が最も小さくなる道を探す問題のこと。交通網の問題に例えれば，2地点間の移動の時間，距離，費用が最も少ないコースを探すことに相当する。

出発点がひとつ，ここからすべての頂点に行ける場合はダイクストラ(Dijkstra)のアルゴリズムがある。出発点に近い頂点から最短路を決定し，2本以上の道があれば短いほうを選ぶ方式である(図はA→Bの例)。



全頂点間の最短路を探る問題にはフロイド(Floyd)のアルゴリズムがある。

最適化(optimization)

①ある系について一定の状況のもとで可能なあり方の中から，最も好ましいものを選び出すこと。あるいは，最も好ましい状態になるように系のパラメタを調整すること。例えば線形計画法などのOR手法で利用する。

②ハードディスク上のデータの記録を整理，再配置して記憶可能な空き領域を増やすディスクの最適化は，デフラグメントという。

最適化コンパイラ(optimizer／optimization compiler)

処理スピードが速く，コンパクトで能率のよいオブジェクトプログラムを生成するコンパイラ。オプティマイザともいう。実行時間の短縮と記憶領域の縮小は必ずしも両立しないが，一般には実行時間の短縮を優先する。

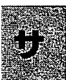
現在，主流のコンパイラはオプション機能として最適化コンパイルを具備しているものが多い。最適化にはプログラムの全般にまたがるグローバル(大域的)な手法と，計算式の変形などのローカル(局所的)な手法とがある。最適化は生成したプログラムの実行時間を短縮することであり，最適化コンパイラのコンパイル時間は普通のコンパイラよりも増加する。

最適性の原理(principle of optimality)

分割統治(divide and conquer／divide and rule)で問題を解決するとき，その手法が正しいことを保証する原理。分割統治とは，問題を部分に分割して解き，それぞれの部分の結果を比較したり，組み合わせて全体の解を得る手法のことである。複数の選択が許される問題で分割統治法が成り立つためには，そのアルゴリズムで最終的に得た解が，他のどの候補解よりも条件にかなっている保証が必要である。この保証がないと分割統治の手法は採用できない。→(参)分割統治

再入可能プログラム(reentrable program)

複数のプログラムが同時に使用できるような構造をしていて，繰り返し実行できるルーチンまたはモジュール。その実行中に自らの内容を変更しないため，先行のタスクが終了する前に次のタスクが利用できる。このようなルーチンやモジュールを再入可能プログラムという。



同時に同一のプログラムを共用することにより，記憶領域の節約や，処理効率の向上を図ることができる。実時間処理で多く用いる。

再配置(defragmentation？／relocation)

フラグメンテーションで主記憶装置上に無駄な空間が多くなったとき，主記憶装置内の再配置可能なプログラムを配置しなおすこと。

再配置可能プログラム(relocatable program)

主記憶装置内のどの領域にロードしても実行可能なプログラム。相対アドレスをもっていて，主記憶に格納する際に絶対アドレスに変換する。プログラムを主記憶装置内の任意の位置に格納できるため，主記憶空間の有効な使用が可能になる。また，主記憶空間を複数の領域に分け，複数のプログラムを記憶させる場合，そのプログラムは再配置可能プログラムであることが望ましい。

再編成(reorganization)

あらかじめ確保したファイルの領域が効率的に使えない時に，ファイルを作り直す処理。索引順次編成ファイルやネットワーク型データベースなどで必要となる。再編成が必要なのは次のような場合である。

(1)領域の無駄使いにより，新しいレコードを追加できなくなった場合
(2)使用できない断片的な記憶空間が増加した場合
(3)一連のレコードが非連続な領域に分布しアクセス効率が低下した場合

また，再編成は一般に次のような手順で行う。

(1)必要なファイルの内容を連続的に配置しながら，他の媒体へ退避する
(2)もとのファイルを削除する
(3)新たに必要十分な領域を確保する
(4)退避していた内容の新たな領域への復元(複写)する

財務システム(financial information system)

広義の財務管理は，資本の調達と運用に関する管理をさすが，狭義には財務会計に関する財務諸表の作成をさす。株主，債権者，取引先，政府機関等に提供する情報である。財務システムは，これらの実施，または意思決定を支援するコンピュータシステム。例えば，経営戦略情報システムや各種の会計情報システムがある。

再利用(reuse)

ソフトウェアの再利用とは，過去に蓄積した知識やプログラムを標準化しておき，これを利用して新たなソフトウェアを創造すること。

再利用技術(reuse technology)

ソフトウェアの再利用のための技術。ソフトウェア生産の効率を高める技術として注目される。大きく次の二つの技術に分けられる。

(1)ソフトウェアそのものを再利用できるモジュールの形にしておき，新たなソフトウェアに組込む。ソフトウェアの開発量を抑制できる(狭義の再利用技術)。

(2)ソフトウェアを作る過程を標準化し，自動化できるツールを開発しておく。ソフトウェアの開発時間を縮小できる(自動化技術)。

先入れ先出し(first in first out)

→(同)FIFO

作業管理(work management)

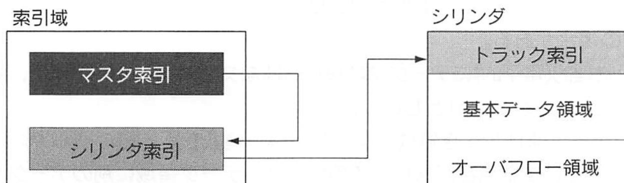
システム開発における人的資源，時間的資源の管理で，進捗管理とほぼ同義。プログラムは目に見えない製品であるだけに，進捗の度合は開発工程を作業単位に分割し，各工程での成果物(設計書や報告書などの文書)の提出，要員の作業報告により把握することが多い。→(参)進捗管理

作業領域(work area)

プログラム実行中にシステムが一時的に使用する記憶領域。通常はシステムが自動的に主記憶，またはカレントディレクトリに確保し，実行後不要になった時点で削除する。

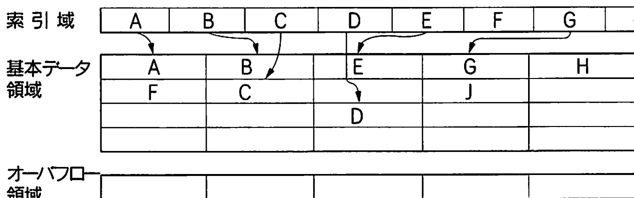
索引域(index area)

索引編成ファイルを構成する3種類の領域のひとつで，実際のレコードの補助記憶装置上の物理的格納場所と，レコード中の項目との対応を示すインデックス(索引)を記録した領域。索引はトラック索引，シリンダ索引，マスタ索引によって構成されるが，索引域に作られるのはこのうち，シリンダ索引とマスタ索引である。



索引順次編成(indexed sequential organization)

乱処理と順次処理の両方を能率よく行えるファイル編成の一方式。ファイル中のレコード内のある項目をキーとし，キーの指定によってアドレスを決め，レコードを読み書きする。単に索引編成ともいう。すなわち，レコードを格納した相対トラック番号と，そのトラック内のキーの最大値とをインデックステーブル(索引)というテーブルで対応づけておく。キーを指定するとそのテーブルを参照し，レコードの呼出しを行う。ブロック化することもでき，その中に空き領域を作っておくことも可能であるが，創成時に入力するレコードは，キーに関して昇順でなければならない。



索引ファイル(indexed file／indexed organization file)

磁気ディスク装置やフロッピーディスク上に作成できるファイル。キー項目の昇順に呼び出す順呼出しと，乱呼出しの双方が可能である。キー項目についてレコードを記憶している場所を示す索引が設定され，呼出しを行うときにこれを参照する。この編成のファイルを扱う場合には，環境部のORGANIZATION句で「INDEXED」指定をする。なお，順ファイルに比べると大きな記憶領域を必要とする。→(同)索引編成ファイル，(参)順ファイル，相対ファイル

索引編成ファイル(indexed organization file)

索引順次編成されたファイル。→(関)索引順次編成

削除(delete／deletion)

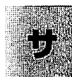
①広義には媒体内のファイルやファイル内のレコードを使用できなくする操作。誤って削除したデータを再度使用可能にすることを復活(または回復)という。削除はOSやアプリケーションソフトによって，次の二つの場合がある。

(1)物理的(電子的)にデータ領域を無意味な文字で書き換える場合，特に消去・抹消(erase)ともいう。

(2)データ領域が空き領域になったという情報を別に書き込んで削除したことにする場合があり，この場合は，データ領域に別のデータがオーバライトされるまでは，アンドゥコマンドやアンデリートコマンドな

どで復活できることがある。→(参)アンデリート，(類)消去

②順編成ファイルではファイルをコピーしながら不要なレコードを取り去り，ファイルを新しく作り直す処理。索引順編成ファイルではレコードの先頭にHIGH-VALUEをもつレコードとして更新すること。索引順次編成ファイルなどでは削除レコードが増加すると格納効率が落ちるため，ときどき再編成する必要がある。



サテライトオフィス(satellite office)

都市機能の集中を解消するために考えだした新しい勤務形態のひとつ。高速で大容量の情報通信路などの情報基盤施設や，各種情報システムを活用することにより，オフィスを都市の郊外に配置する形態，またはその高度情報化オフィス。都市機能の集中を解消するために考え出された新しい勤務形態のひとつでもある。本社とはISDNなどの高度情報通信回線で結び各種のOA機器を設置して，データベースの相互アクセス，電子メール，電子会議システムなどを活用して職務遂行の便宜を図る。社員がそこに通って仕事をすることで，郊外の住宅地から大都市中心部の会社への長い通勤時間から解放される。またオフィススペースの確保も容易となる。その結果，マクロな経済効率向上が期待できるなど利点がある。反面，会社との一体感が薄れる，同僚などとの意思疎通が図りにくい，人事管理が困難などの問題点もある。→(関)SOHO

サブシステム(subsystem)

大きな統一的なシステムの一部分の機能を分担するシステム。大きく複雑な機械でも，いくつかの機能をもつ部分から構成されるように，コンピュータシステムもより小さな複数のシステムから構成することができる。その小さなシステムもさらに小さなシステムの集合ととらえることができる。このような階層的な見方で，上位のシステムから下位のシステムを見るとき，サブシステムという。

サブスキーマ(subschema)

CODASYL型データベースを応用プログラムから見たときの，処理すべきデータおよびその論理構造。応用プログラムからデータ操作の呼出し定義時に指定(記述)する。サブスキーマ言語により，使用可能なデータベースの部分の論理的な構造，レコードの使用可能な構成要素，親子関係の使用可能な型を定義する。サブスキーマが応用プログラムから扱えるデータを制限するため，データの機密保護が確保できる。

→(関)スキーマ，データベース

サ

サブディレクトリ(subdirectory)

MS-DOSやUNIX系OSなどの階層構造のファイル管理システムで，あるディレクトリのすぐ下位にあるすべてのディレクトリ。分岐ディレクトリともいう。サブディレクトリは，さらにその下位に複数のサブディレクトリをもつことができる。これらは独立のディレクトリ名をもつことができ，主に数多くのファイルのグループ化に利用する。あるサブディレクトリから見て，自分のすぐ上位にあるディレクトリを親ディレクトリという。

サブメニュー(submenu)

階層構造をもつメニュー選択法のひとつ。あるメニューのすぐ下位にあるメニュー。多機能ソフトウェアのメニュー表示では，全機能を数グループに分け，代表としてメニューバーやメインメニューに常時表示し，これを選択したとき，さらにグループ内のメニューを表示する。この下位のメニューをサブメニューという。サブメニューを選択すると，さらにその下位のメニューを表示する場合もある。

→(関)ポップアップメニュー

サブルーチン(subroutine)

プログラムの論理的にまとまった機能部分で，必要な箇所で呼び出して使用するルーチン。処理の流れの中で共通機能を多数回利用する場合，その機能をサブルーチンにすることが望ましい。プログラムの記憶域，作成時間，労力が節約できるため頻繁に利用する手法である。

サブルーチンコール(subroutine call)

プログラムからサブルーチンを呼び出すこと，またはそのための命令や文。サブルーチンは複数のプログラムから何回でも呼び出すことができるので，システムの機能分割に活用できる。また，プログラム言語に依存せず，他のプログラムからでも呼び出して利用できるものもある。

差分ファイル(differential file)

ソフトウェアのバージョンアップやデータのバックアップの際に，以前のバージョンから変更になった部分を記録したファイル。ソフトウェアのバージョンアップを行う際に旧版ユーザーが新規ユーザーより安価に新バージョンを使用できるよう，バージョンアップパッケージ(アップグレードパッケージともいう)を提供することが多い。バージョンアップパッケージには以前のバージョンとの差分ファイルを格納しており，以前のバージョンのソフトウェアが存在しなければ，新バージョンのソフトウェアをインストールすることができないということも多い。

サポート(support／user support)

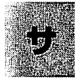
①コンピュータシステムなどにおいて，高度な専門知識を必要とするハードウェアやソフトウェアを維持，保守する体制。

(1)企業システムでは，開発部門，開発会社がユーザ部門の教育，指導，保守，リカバリなどに当たるケースがある。

(2)パソコンシステムでは，ハードメーカ，ソフトメーカの専任のアドバイザがユーザへの助言やアフターサービスに当たるケースがある。

サポートはその内容によって無料と有料の場合がある。なお，システムの正規のユーザとして登録した場合に限られることが多い。

②あるソフトウェア製品が別のソフトウェアと矛盾なく共存できたり，機能を提供できること。例えば，アドインソフト，アドオンソフトや周辺機器の機能拡張ソフト(A)が，既存ソフト(B)の機能を拡張するとき「AはBをサポートする」などという。



サポートセンタ(user support center)

ハードウェアやソフトウェアのトラブル，あるいは操作する上でわからないことなど，ユーザからの問い合わせに対応する，メーカまたはメーカ委託のサービス窓口。パソコンが一般に普及するにともない，各社ともサービスの一環として体制を強化している。急増する問い合わせに対応しきれず，どのメーカのサポートセンタも要員不足状態にあるのが実情。また，コンピュータはハードウェア，OS，アプリケーションなど，さまざまな要件がからみあい問題点を切り分けるのが困難で，たらい回しにされることもあり，ユーザ側の不満も高まってきた。こうしたことから，これまで一般ユーザに対するサポートを原則的に無料で行ってきた方針を見直し，より充実したサービスを提供するために有料化へと移行するメーカが増えてきた。

サムチェック(sum check)

→(同)合計検査

産業用ロボット(industrial robots)

主として工場内で熔接，組立，運搬等の作業等を行うロボットで，メカトロニクス技術とコンピュータ技術の結合により，高い機能を有するもの。自由度の高い多様な動作が可能で，単なる自動機械とは区別することがある。JISでは一般的分類として，下の 5つをあげている。

(1)操縦ロボット(人間が直接操作する)

(2)シーケンスロボット(あらかじめ設定された順序で動作を逐次進める)

(3)プレイバックロボット(人間がロボットを動かして，動きを教示する)

(4)数値制御ロボット(動きの情報を数値，言語などにより教示する)
(5)知能ロボット(人工知能で行動を決定する)
・感覚制御ロボット(感覚情報により動作を制御する)
・適応制御ロボット(適応制御機能をもつ)
・学習制御ロボット(学習制御機能をもつ)

参照(reference)

①プログラムで，対象の名前を使用すること。引用ともいう。例えば，変数名や定数名を式の中で指定すること，配列要素の位置を指標や添字で指定すること，プログラム中で関数名を使用すること。参照により，構成要素をパラメタで一意に指定したり，同一手続きが利用でき，プログラムを単純化できる。参照方法には，変数名や定数名などの名前指定，配列要素の位置を指標や添字による指定，データを代表するコードによる指定などがある。

②データベースやファイル中の指定したデータを得ること。

サンプリング(sampling)

測定や統計調査の目的で，母集団から標本や試料(サンプル)を抜き取ること。母集団が個で構成されている場合は抜取りともいう。

(1)全数調査では時間や費用がかかるとき，統計の正確度をある程度犠牲にして，一部を調査して早く結果を知りたいときにサンプリングを行う。無作為抽出，典型抽出などの手法がある。

(2)製品の品質管理では，全数調査ができない場合や抜取り調査で十分な場合，一定のサンプリング間隔で，または規定のサンプリング単位で抽出する。

シ

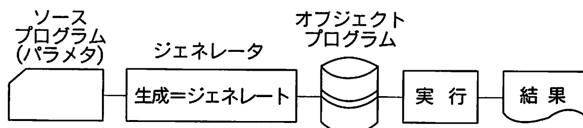
シームレス化(aim of seamless)

企業内システムで，部門間でのデータの受け渡しや処理を連続的にスムーズに行うこと。一般に，部門が異なるとき生じやすいデータの再入力や処理の分断を抑制すること。一連の業務が多部門にわたるとき，データ処理の流れの中で部門ごとにデータを転記したり，再入力したりする手間を要さないように，電子化データをコンピュータ間で受け渡し，流れを分断させないこと，または，データ処理の流れを連続的に行うこと。これを情

報だけでなく，サインや印鑑に代わる電子認証技術を利用した書類の流れなどにも適用して業務を効率化しようという傾向にある。

ジェネレータ(generator)

あらかじめプログラムの骨組みができており，利用者が入力データや処理結果の内容と形式，および処理条件などを一定の書式の各欄に記入，入力すると，自動的に処理に必要なプログラムを作成してくれるプログラム。生成プログラムともいう。代表的なジェネレータにRPGがある。



シェルソート(Shell's sort／diminishing increment sort)

内部ソートのアルゴリズムのひとつ。基本的な考えは，挿入ソートが大半のデータが正順で一部データが乱順である場合に効率がよい性質を利用し，まず大ざっぱに整列し，最後に挿入ソートで仕上げる方式である。

配列中のデータから，まず一定間隔(例えばm)ごとにデータをとって整列する。これをずらしながらm回繰り返す。全データをmグループに分け整列したわけで，大まかには整列される。間隔幅を変えて大まかな整列を繰り返せば，整列の状態に近づく。そこで挿入ソートを行えば，効率よくソートが完了するというアルゴリズムである。実験的には，データ量の1.25乗に比例する平均計算量が必要であるという。

時間計算量(time complexity)

アルゴリズムの効率を評価する目安のひとつで，処理の終了までに要する実行時間の長さ。一般的にはハードウェアの速度に依存する絶対的な処理時間ではなく，実行ステップの回数やデータがn倍に増えた場合のステップ数の増加の割合で議論する。同じアルゴリズムであっても，入力した値によって手間が異なる。そのアルゴリズムにとって最も時間を要する値を与えたときの計算量を最大時間計算量(worst case time complexity)といい，考えられるすべての入力の計算量の平均を平均時間計算量(average time complexity)といい，計算量を示す目安とする。→(参)オーダ

式(expression)

プログラムを記述する基本要素で，定数や変数，関数などの演算要素を演算対象として演算子で結合したもの。表計算ソフトやデータベースソフトでは，データ要素間の演算式やデータの条件を指定する条件式，論理式などをさす。式は，式自体を演算対象とすることができ，複雑な手順の演

算もまとめて記述できる。式のもつ値や演算子の種類により，算術式，論理式などに分かれる。式の値を得ることを式の評価という。

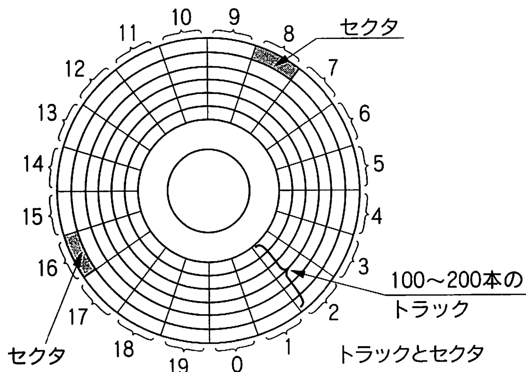
磁気ディスク(magnetic disk)

磁性体を両面に塗った回転円盤で，その表面を磁化しデータを記憶する媒体。次のような特色がある。

(1)回転盤上のどの位置にでも直接データを記録することができ，また記録したデータを読み出すことができるため，磁気テープに比べ多様な処理が可能になる。

(2)円盤の素材が金属，ガラスなど剛体であるものをハードディスク，合成樹脂の柔らかい種類をフロッピーディスク，またはフレキシブルディスクという。

(3)高速回転する磁気ディスクに，アクセスアーム，磁気ヘッドと組み合わせて記憶を行わせる装置を磁気ディスク記憶装置という。



トラックとセクタ

→(例)フロッピーディスク，ハードディスク，(関)セクタ，クラスタ

磁気テープ(magnetic tape：MT)

プラスチックをベースとしたテープに磁性材料を塗布した記憶媒体。磁化の方向によってデータを記憶する。長いテープを巻いて使うため逐次呼出しとなって呼出し時間がかかるため，パソコンの分野ではバックアップ専用の記憶媒体として用いる。汎用コンピュータ利用分野では，データの長期保存やコンピュータどうしの情報交換に適し，何回でも書き換えることができるため，重要な記憶媒体となっている。表面の欠陥，異物の付着で信号の誤り(ドロップアウト)が生じる欠点がある。そこでオープンリール型の磁気テープに代わり，最近はケースに収納したまま扱えるカートリッジ式の磁気テープが主流となった。

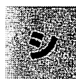
磁気テープラベル(magnetic tape label)

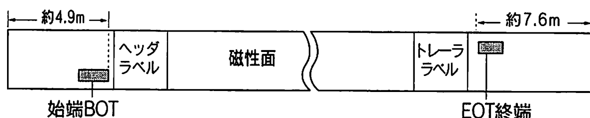
磁気テープにおいてボリュームやファイルの識別に使うラベル(標識名など)のこと。

(1)ボリュームの前後に書くVOLラベルとEOVラベル，ファイルの前後に書く見出しラベルと後書きラベルの4種類がある。

(2)ラベルとファイルレコード間の区切りやファイルの最後にはテープマーカ(tape marker：TM)という1バイトの特別な符号を自動的に記録する。

(3)データの記録可能な範囲を示すための二つの反射マーカがついており，始端を示すマークをBOT(beginning of tape)，終端を示すマーカをEOT(end of tape)という。





識別子(identifier：ID)

広義には，文字や数字を，一意に識別できるように組み合わせた文字列。プログラム中でのファイル名，変数名，定数名，装置名，段落名など，それぞれの範囲内で同一の文字列であってはならない。

字句(lexical unit)

プログラムを構成する意味をもつ言語要素。予約語，変数，定数，区切り符号，演算子，行番号，注釈など。

字句解析(lexical analysis)

翻訳プログラムがソースプログラム中の字句を取り出し，分析すること。変数や配列の名前などは型や長さなどの属性を判断し，構文解析のプログラムに引き渡す。→(参)コンパイル，意味解析，構文解析

時系列分析(time series analysis)

時系列データの構造はどうなっているかを分析すること。時系列データを傾向変動，循環変動，季節変動，不規則変動に分けるのが普通であるが，これらの変動はあくまでも仮定のものであって実在するか否かはわからない。乗法モデルを用いて分析する方法と，加法モデルを用いて分析する方法とがある。

試験(①test，②check)

①作成したシステムやプログラムの正当性を検査したり，性能を測定すること。→(同)テスト

②入力したデータの正当性の検査。検査，チェックということが多い。

資源管理プログラム(resource management program)

コンピュータのシステム資源を効率よく管理するためのプログラム。限られた資源を効率よく使用し，スループットの向上やコンピュータシステム利用コストの低減を図る。コンピュータシステムで，CPU，メモリや磁気ディスク記憶装置等の各種入出力装置をハードウェア資源，プログラム等をソフトウェア資源という。オペレーティングシステムの役割のひとつにこれら資源の割当てや管理がある。

自己展開形式(self-extracting form)

実行可能なプログラムを圧縮したファイル形式のひとつ。実行コマンドを受けると自動的にファイルを展開し，実行するプログラムを付加したもの。解凍用のプログラムを利用する必要がなく，パソコン通信などでプログラムを授受するのに便利である。MS-DOSやWindowsでのファイル拡張子は，見かけは実行可能ファイル「.EXE」であるが，実行するとディスクにファイルを展開するため占有容量が増える。Macintoshの場合は「.sea」が自己展開形式ファイルの拡張子である。LHAでは，自己解凍形式という。

辞書(dictionary)

一般には日本語ワードプロセッサで用いる，ローマ字かな変換辞書，かな漢字変換のための熟語辞書。広義には，関連づけたコードや情報等を一定の規則に従って参照できる形に集めたファイル。

システムアドミニストレータ(system administrator)

情報システムの利用者の立場で，エンドユーザコンピューティング(EUC)の推進に従事し，EUC環境の充実とツールの利用を推進する技術者。利用者側において情報処理に関する一定の知識，技術をもち，部門内あるいはグループ内の情報化を利用者としての立場から推進する。具体的には，以下のような役割をになう。

(1)エンドユーザコンピューティングの推進
(2)部門内あるいはグループ内の情報システムの構築またはその支援
(3)情報システムの提供者側に対する利用者側の意見要望の提起
(4)情報システムの運用とシステム利用環境の整備

システムアナリスト(system analyst)

情報システム化の全体的な企画にかかわり，情報戦略の立案，中・長期の情報システムの全体計画，個別情報システムの開発計画を立案し，システム化計画書の作成を行う技術者。開発計画が関係者の承認を得たら，プロジェクトマネージャなどの開発担当者に業務を引き継ぐ。

システムインテグレーション(system integration：SI)

システムの高度化，多様化，複合化要求にともない，使用中のシステム資源全体を見直し，これらを統合的に組合せ，新しい要求に対応できるように最適なシステムを構築，保守，管理すること。特にOA化の進展の中でセンタコンピュータ(ホストコンピュータ)，ワークステーション，オフコン等を接続するオンライン処理要求の増大に対して長期的な視点からシステム構築を考える手法。

システム運用(system operation)

ユーザによるシステムの稼働，システム資源の管理，システムの機能，性能を継続的に維持する作業。広義には運用設計，運用基準の策定，サービスの提供を含む。一般的な運用管理は，次のような領域で考える。

(1)性能管理：規定の処理能力の維持・性能の測定や評価
(2)障害管理：障害の監視・検知・原因や影響の把握・回復や保守
(3)運用管理：運用の合理化や自動化・運用経費の算出や課金
(4)機密管理：システムの情報資源の悪用や盗用の防止

システムエンジニア(systems engineer：SE)

システムを構築するとき，要求定義，現行システムの調査分析，新システムの概要設計，詳細設計までの職務を担当する技術者の総称。要求定義の作成に従事する人をシステムアナリスト，システム設計に従事する人をシステムデザイナと分ける場合もある。通産省の指導で作成した高度情報化人材育成標準カリキュラムは，さらに専門分野によって細分化している。

システムオペレータ(system operator)

広義には，コンピュータシステムの操作員。コンピュータを起動したり，プログラムを実行したり，媒体を補助記憶装置にセットしたり，計算結果を取り出すなどの作業を行う。単にオペレータともいう。

システム開発(system development)

ユーザの要求を把握し，システム設計，プログラム設計，プログラム開発，システム運用までの工程を管理し，実現すること。大規模なプログラム開発と同義に用いる。総合的，広域的にわたる開発をさすことが多い。
→(類)プログラム開発，(類)システム設計

システム開発ライフサイクル(system development life cycle：SDLC)

システム開発の過程を，ユーザの状況の変化に対応して連続的，周期的に生ずるものという観点でとらえる場合の，システム開発の工程。段階別分業化の考えを取り入れ，各段階の作業量，技術水準を考慮した人材，時間，費用を投入し，確実に効率よくシステムを開発するために考えられた。

シ

システム監査技術者(system auditor)

客観的な第三者の立場でコンピュータ情報処理システムの安全性，信頼性，有効性などを総合的に点検，評価し，関係者に助言や勧告をする技術者。システムの有効な活用とセキュリティ対策の実効性向上などを企図する。情報システム部門および利用部門が，いかに情報システムの信頼性，安全性，効率性の向上に気を配って業務を遂行したとしても，情報システムに関連して多種多様の問題が生ずる。特に，コンピュータおよびその他情報機器の高性能化，通信ネットワークの進展にともない，ますます情報システムが大規模化，高度化して，当事者の努力だけでは，問題事項を解決することが困難になってきている。日本では公認会計士や内部監査人が実施する場合が多く，専任のシステム監査人は少ない。

システム監査基準(system auditing)

1985年 1月に通商産業省が，情報処理システムの安全性，信頼性，有効性などを総合的に点検，評価し，関係者の助言や勧告をすること，システムの有効な活用とセキュリティ対策の実効性向上などを目的として，公示した基準。→(参)システム監査技術者，情報システム安全対策基準

システム管理者(system manager)

コンピュータシステムを運用，管理する人間。資源管理，運用時間管理，ユーザ登録管理，課金管理，ネットワークやファイルの障害管理，セキュリティ管理，性能管理など，さまざまな管理を行う。インターネットへの接続が普及するなど，ネットワーク接続先が増加するとシステム管理者の責任と役割はさらに大きくなる。

比較的大規模な分散処理システムではコンピュータシステムが各所に分散する。そこでは，分散処理システム全体を管理するシステム管理者とローカルなシステムを管理するシステム管理者が必要となる。それぞれ重点を置く役割は異なるが，相互に綿密な連絡を取り合いながらシステム全体の維持を行う。

システム構成(system configuration)

コンピュータシステムでは，中央処理装置，主記憶装置，外部記憶装置，通信制御装置などの組合せを意味する。

→(例)シンプレックス構成システム

システムコール(system call)

利用者やシステム内のプログラムに対して，オペレーティングシステムが提供するサービス機能の呼出し。利用者はファイルアクセス制御やプログラム管理などの処理を自分で複雑なプログラムを作成することなく，シ

ステムコールにより行うことができる。→(参)オペレーティングシステム

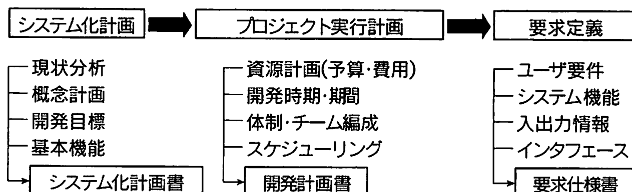
システム資源(resource)

コンピュータで処理を行う際に使用するハードウェア，ソフトウェアの時間や容量，労力，エネルギー，資材等の総称。オペレーティングシステムがこれらの資源を管理し，使用可能な主記憶領域や出力装置，磁気ディスク記憶装置の残り容量等をオペレータに示す。

マルチプログラミングシステムでは，これらの資源を最も有効に利用すべくジョブ管理を行っている。→(参)資源管理プログラム

システム設計(systems design)

コンピュータシステムの目的，機能，性能を設定し，実現するための具体的方法を設計すること。ハードウェア，ソフトウェア，時間，要員等の資源をどう配分し，構築し，運用するか，方針を決定する。システム設計の手順は開発するシステムの特性や規模によって異なる。基本的には，新システムの目標の設定，外部設計，内部設計などプログラム開発に先立つ上流工程の基本計画，設計作業をさす。



システムセットアップ(system setup)

新規のハードウェア構成要素を互いに接続し，適切な設定を行い，記憶媒体の初期化，基本ソフトウェアのインストール，アプリケーションソフトのカスタマイズなどを行う作業。次のような作業の全体を指す。

(1)各装置に必要な部品要素(本体，インタフェース，ケーブルなど)やソフトウェア(デバイスドライバ，組込み用ソフト，マニュアル，利用者登録表など)の確認
(2)増設メモリや増設インタフェースボードなどの本体内部への取り付け
(3)周辺機器と本体のケーブルの接続，各種スイッチの設定，電源の投入
(4)オペレーティングシステムのインストール
(5)増設装置のドライバのインストール。外部記憶媒体の初期化など
(6)アプリケーションプログラムのインストール，カスタマイズ
(7)コンピュータシステム全体のシステム環境設定

システムソフトウェア(system software)

コンピュータシステムの一般の利用者のために，あらかじめ用意してあるプログラムの総称。その役割から次の二種類に分類できる。

(1)基本ソフトウェア：広義のオペレーティングシステムとほぼ同義。入出力や実行制御，資源を確保する制御プログラム，コンパイラなどの言語プロセッサ，プログラム支援ツールや各種標準ソフトウェアパッケージなどのユーティリティプログラムなど

(2)ミドルウェア：応用ソフトに基本的機能を提供するソフトウェア。通信管理，GUI制御，グラフィック処理プログラムなど

システムディスク(system disk)

コンピュータを運用するためのシステムプログラム(オペレーティングシステム，基本ソフトウェア)が存在するハードディスクやフロッピーディスク。システムを起動するプログラムである。パソコンを起動するのに必要なシステムソフト(例えば，MS-DOSやアプリケーションの本体やサービスプログラム)を記憶しているフロッピーディスク，CD-ROM。これに対し，主としてユーザが作成したプログラムやデータを記録するディスクをデータディスク，ユーザディスクといい，コンピュータを起動できない。

システムテスト(system test)

開発されたシステムが，総合的に見てシステム設計段階の目的，目標を実現しているか確認するテスト。総合テスト(integrated test)ともいう。おもなテスト項目は，機能テスト，過負荷テスト，安定性テスト，耐障害テスト，性能テストなどであり，運用の現場にシステムが移行された場合を想定して開発者が行うテストである。オンラインシステム，バッチシステムの両方で構成されたシステムでは，資源の競合が相互に影響を及ぼすので，実際の運用に即したテストが必要である。

→(参)受入れテスト，結合テスト，モジュールテスト

システム流れ図(system flowchart)

システム全体にわたるデータの流れと処理の内容を示す図面。原始データが各種の処理を受けて最終的な出力情報になるまでの順を図示する。この流れ図により，データを処理する各過程でどのような資源が必要になり，どのような処理を行うかを一覧することができる。

→(参)流れ図，流れ図記号

システムファイル(system file)

ファイルをその利用主体により分類した場合の用語で，システム自体が使用するファイル。次のようなものが該当する。狭義には(1)をさす。

(1)制御プログラム，サービスプログラム，言語翻訳プログラムなどを格納するファイル
(2)ユーザの応用プログラムを格納するファイル
(3)システムプログラムの実行時に使用する作業用ファイル

システムプログラム(system program)

コンピュータシステムの一般の利用者のために，あらかじめ用意してあるプログラムの総称。利用者のプログラムの入出力や実行制御，資源の確保からなる制御プログラム，コンパイラなどの言語処理プログラム，プログラム支援ツールや各種標準ソフトウェアパッケージなどのユーティリティプログラムをさす。オペレーティングシステムとほぼ同義。

システム分析(system analysis)

情報システムの構築に先だって，経営課題，事業環境，業務改善策などを考慮しながら，システム化計画を検討すること。現状分析，要求分析，情報システムの定義，日程，コストなどのデータから，システム開発計画を練る工程。

自然言語(natural language)

日常使用している日本語や英語などの言語。プログラム言語(COBOL，FORTRAN等)などの人工言語に対して自然言語という。ヒューマンインタフェースの改善という観点からはプログラム言語も自然言語に近づけたほうが好ましい。しかし，自然言語のあいまい性(解釈に依存する部分)などが開発の障害になり，自然言語でコンピュータに命令を与えるまでには至っていない。→(対)人工言語，(参)機械翻訳

シソーラス(thesaurus)

データベース検索や文字列検索に使用する類似語，類義語を整理した辞書のこと。シソーラスではキーワードとその関係(同義語，類義語，広義語，狭義語，上意語，下意語，反意語，異表現，関連語など)を規定して配列する。機械翻訳や情報検索の分野で，語群の統一や情報検索のキーワードの融通性として重要な概念である。コンピュータで自然言語による情報検索をおこなう際にユーザーの要求に該当する必要十分な情報を得るためには，シソーラスを参照したキーワードを使って効率のよい論理演算式を作る必要がある。

自治体情報提供システム(local government information providing system)

地方自治体(県，市町村など)の行政機関が扱う情報を提供するシステム。現在は，各種統計，行政サービス(市民施設の利用，自治体経営住宅)など限られた生活情報の範囲で提供している。

実現値(occurrence／instance)

個々の実体の属性を表現する値のこと。データベースのレコードの値そのものをさす。実現値は，(実体の数×属性の数)だけ存在する。

実行(execution)

コンピュータシステムによって一連のコマンド，またはプログラム中の複数の命令や文を自動的に遂行する処理。広く，コマンドを解釈して処理することから，プログラムによる処理結果を得ることまで含む。

実行可能プログラム(executable program)

コンパイル，リンクを行い，コンピュータを動作させる準備がすべて完了したプログラム。ロードモジュールはこれに相当する。

実行文(executable statement)

プログラムの主要な構成要素で，データに一定の処理を施したり，実行順序を制御する。前者には，代入文，入出力文などがあり，後者には，繰返し文，条件文などがある。→(対)非実行文

シノニム(synonym)

直接編成ファイルでファイル中の欠番キーが多く，直接アドレスによる方法が不向きな場合，アドレス生成法(除算法，数字分析法，重ね合わせ法，基数変換法等)のハッシュ関数を用いて実際に格納するアドレスの範囲を圧縮する。このとき複数のレコードの異なったキーを同一のアドレスに変換してしまうことがあり，これをシノニムという。

キー項目を変換して得たアドレス(本来のアドレス)をホームアドレスといい，ここに入るレコードをホームレコードという。ホームアドレスをすでに他のレコードが占めているとき，シノニムレコード(あふれたレコード)を格納するアドレスをシノニムアドレスという。→(参)ハッシング

指標(index)

①命令のアドレス部に適用する修飾子。→(同)インデックス

②COBOLで，表操作に用いる添字。

シミュレーション(simulation)

複雑なシステムや現象を数学的，物理学的，工学的にモデル化してコンピュータに計算させた結果を使って，それらの現象を解析する手法。模擬実験のこと。コンピュータを用いないシミュレーションの考え方は古くからあり，手作業でもできるが，モデルに対し種々の変数を与え多量の計算をする必要があるため，コンピュータとともに発達した。乱数を発生させ解析するシミュレーションにモンテカルロ法がある。

シミュレーションソフト(simulation software)

さまざまな現象をコンピュータ上で模倣し，各種の実験や訓練を行うソフトウェア。例えば，実際に自動車を破壊することなく衝突時の安全性を検証したり，宇宙空間での仮想のトレーニングに活用したり，あるいは株価の変動のような経済予測にも用いられている。パソコン分野での代表的なシミュレーションソフトはゲームである。戦闘機の操縦を疑似体験するフライトシミュレータなどがある。



シミュレータ(simulator)

物理的な現象または抽象的なシステムのふるまいをモデル化し，模擬実験を行うためのハードウェアまたはソフトウェアのシステム。

事務管理(office management)

企業内の業務にともなう書記的な作業を管理すること。企業が発展すると事務量が増大し，取り扱い品目が多様化するとさらに増大する。また企業の社会的責任が重くなると，事務作業は重要度を増す。これらの複雑，多岐にわたる文書処理を科学的に分析，管理すること。

事務データ処理(business data processing)

事務に関わるコンピュータ利用のデータ処理。科学技術計算に対比して用いる。科学技術計算に比べ，1件当たりの計算量は少ないが，件数が著しく多いという特徴がある。→(対)科学技術計算

事務分析(office analysis／procedure analysis)

企業内の事務作業を効率化，合理化する目的で，科学的な立場から，業務の分析を行うこと。事務手続き分析，事務工程分析，事務時系列分析ともいう。事務分析では，次のような点に関して調査を行う。

(1)事務がどのような手続きですすめられているか
(2)事務量やデータ量はどのくらいあるか
(3)よく起こる誤りには，どのようなものがあるか
(4)事務担当者が，どのような改善を希望しているか

射影(projection)

関係データベースのデータ操作の一機能。元になる表の中の特定の項目(属性)だけを指定して，それだけの新しい表を作る機能。
→(関)関係代数，(参)関係データベース，(関)選択，結合

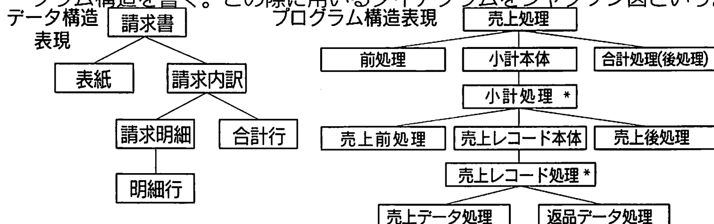
社会システム(social life system)

社会制度，社会生活にかかわりの深いコンピュータシステムの総称。おもに，公共企業(金融，交通，エネルギー関連企業)の提供するシステム，公共団体(政府，自治体など)のシステムをさす。市民生活に影響が大き

く，大規模なネットワークとデータベースに支えられている例が多い。

ジャクソン法(Jackson structured programming)

1970年代半ばに Michael Jackson によって提案されたデータ構造と処理構造の図式的表現法。データ構造からプログラム構造(処理構造)が決まるとしている点で，データ本位の構造化手法である。このために4つの基本図式を用い，入力データと出力データの構造を書き，この関係からプログラム構造を書く。この際に用いるダイアグラムをジャクソン図という。



ジャンプ(jump)

制御をプログラムの別の場所へ無条件に移すこと。ジャンプ文にはgoto文(無条件分岐)，break(ループ，スイッチからの脱出)，continue(ループの終了地点への分岐)，return(関数から呼出し元への復帰)などの文がある。プログラムによって異なるが，これらのジャンプ文は関数内，ブロック内での分岐に限られる場合がある。

集計検査(sum check)

→(類)合計検査，件数検査

集合演算(set operation)

関係データベースの二つ以上の表に対する操作は，演算として定義できるが，この演算の集まりをいう。複数の表から新しい関係を作ることができる。次の3種類の基本演算がある。

(1)複数の表のすべての要素を集める和演算(union operation)
(2)複数の表の共通する要素を集める積演算(intersection operation)
(3)ある表の要素から別の表の要素を除いた要素を集める差演算(difference operation)

集合操作(set operation)

SQLのDMLで，ひとつの命令で複数のレコードをまとめた集合(set)を処理できる命令群(SELECT，INSERT，DELETEなど)がある。この命令による複数レコードの操作のこと。

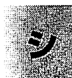
修正作業(debug／correct)

処理，手続きなど仕様との隔たり，プログラムの誤りなどを修正する作

業。運用段階に入っての修正は，最も緊急の作業である。修正後は改めてテストを行い，修正の影響が他のプログラムに及んでいないことを確認することが重要である。

従属セグメント(slave segment)

→(同)処理・加工セグメント，(対)制御セグメント



従属モジュール(slave module)

プログラムの構造化設計でモジュール分割したとき上位のモジュールから呼び出されるモジュール。下位モジュールともいう。

終値(terminal parameter／terminal value)

繰返しループを実行するに先だって，制御変数，領域に設定する定数，条件の値の一つ。制御変数の値がこの値を越えると繰返し処理が終わる。

集中データ処理(integrated data processing：IDP)

センタの大型のホストコンピュータと複数の端末を通信回線で接続し，業務で発生するデータをすべて，センタのホストコンピュータに集中させ，一括処理し，結果を各端末に返送する処理方式。集中データ処理を行うコンピュータシステムを集中データ処理システム(IDPS)という。プログラムやデータが集中しておりシステムの保守性はよい。しかし，ターンアラウンドが悪いこと，障害復旧処理に時間がかかる欠点がある。これとは反対に，データや処理機能を分散して全体として効率的な処理を行うことを目指したのが分散データ処理である。→(対)分散データ処理

集中データ処理システム(integrated data processing system：IDPS)

集中データ処理を行うコンピュータシステム。

出力(output)

コンピュータで処理した結果を外に出すこと。例えば，ディスプレイ画面への表示，プリンタへの印字，補助記憶装置への書込み，通信回線への送信などがある。

出力イメージ(output image／output design)

出力設計の工程で作成する出力画面や出力帳票のレイアウト情報。出力帳票や印刷文書の体裁(見出し，罫線，用紙サイズ，文字の書体，大きさ，金額表示のルール)等である。いずれもシステムの目的にあわせて決定する。→(参)画面設計，報告書設計，帳票設計

出力機能(output function)

コンピュータシステムがレジスタや主記憶装置に記憶しているデータを，磁気ディスクなどの外部記憶装置に書き込み，ディスプレイ画面に表示し，プリンタなどに送り出す機能。

出力設計(output design)

コンピュータシステムが，いつ，だれに，どのような情報をどんな媒体で出力するのか設計する工程。データ処理の出力では，分かりやすく，必要十分な情報をタイミングよく提供する出力設計が重要になる。具体的な設計の範囲は，出力機器，媒体の選定，印字，表示のレイアウトから出力の配布，配送法まで多岐にわたる。



→(参)帳票設計，スペーシングチャート

出力センター(output center／service bureau)

高解像度・高品質の出力装置(イメージセッタ)を用いて，DTPソフト，グラフィックソフトで作成したドキュメントやページをフィルムや印画紙に出力するサービス業者。

出力帳票(output／output sheet／report)

人間の読める形式に，データを印字したもの。その利用目的，印字装置の制約などを考え，用紙サイズ，レイアウト，プレプリントの有無，フォントなどをくふうする。出力帳票は次の点に注意して設計する。

(1)必要十分なデータが用紙に納まること

(2)出力項目がユーザにわかりやすい体裁であること

(3)関連帳票との記入位置，記入項目を統一すること

(4)ファイリングしやすい形であること

出力媒体(output data medium)

コンピュータの処理結果を出力装置により記録する媒体。人間が読みうる形式の媒体と，記憶媒体上に記録して別の出力装置により人間が読みうる形にする媒体がある。前者の例には，ディスプレイ装置，出力帳票が，後者の例には，磁気ディスク，フロッピーディスクなどがある。

出力レイアウト(output layout／output image／output design)

ディスプレイ画面や出力帳票のレイアウト情報。特に出力帳票や印刷文書の体裁をさす。見出し，罫線，用紙サイズ，文字の書体，大きさ，金額表示のルール等である。いずれもシステムの目的に合わせて決定する。また，出力の配布，配送法まで考慮する。

受発注管理(ordering management)

商品の受注，発注に関する情報の管理。販売管理，在庫管理と関連が深い。EOS(electronic ordering management)は，受発注業務の効率化，自動化を目指すと同時に，時間短縮を可能にする。→(関)EOS

需要予測(demand forecasting)

需要量が将来どのようになるかを予測すること。過去のデータの引延し

的な方法としては，時系列データに曲線を当てはめて外挿により予測する方法，季節変動指数を求めて予測する方法，指数平滑法，横断的データによる方法，ロジスティック曲線やゴンペルツ曲線による方法など種々の手法が考えられている。



順次アクセス(sequential access)

あらかじめ定められた順序(例えば発生順，記録順)に従い，順番にアクセス，処理していく方法。磁気テープ上のファイルに対するアクセスなど，全レコードを順番に処理する場合に効率のよいアクセス方法である。特定のレコードにアクセスする場合もその前に位置するレコードをすべて読む必要がある。→(対)直接アクセス

順次アクセス記憶装置(sequential access storage device)

アクセスに要する時間が，データの記憶領域とその直前に参照するデータに依存する記憶装置。データのアクセスを順次(例えば発生順，記録順)に行う記憶装置で，例えば磁気テープ記憶装置がある。これに対比されるのが，直接アクセス記憶装置(DASD)である。

順次アクセスファイル(sequential access file)

順次編成ファイル，索引順次編成ファイルのように，順次アクセスに適したファイル。→(対)直接アクセスファイル

順次処理(sequential processing)

あらかじめ定められた順序で順番に処理していくこと。ひとつの処理が終わってから次の処理を行うため，処理に時間がかかるが，同時に二つの部分を処理しないため，誤りは少ない。逐次処理ともいう。

順番検査(sequence check)

ファイルのレコードが，特定のキーを基準として，正しい順序関係に保持されているか確認すること。特にデータを複数の媒体で保持している場合，コンピュータで処理する前に並び順の検査が必要である。

順番コード(sequence code)

一連番号をそのまま利用したコード。単純で最も少ない桁数でコード化できる。いったん決めるとコードの追加は末尾へなら可能であるが，挿入はコード変更をともなうため，永続性のある半永久的な固定したデータ，単なる受け付け順連番などに用いることが多い。

〈例〉JIS都道府県コード

01北海道，02青森県，03岩手県，……，13東京都，14神奈川県，
15新潟県，……，46鹿児島県，47沖縄県

順ファイル(sequential file)

磁気テープ上に作成したファイルのように，記憶した順番にしか呼び出すことのできないファイル。磁気ディスク装置などにも作成可能である。

順編成(sequential organization)

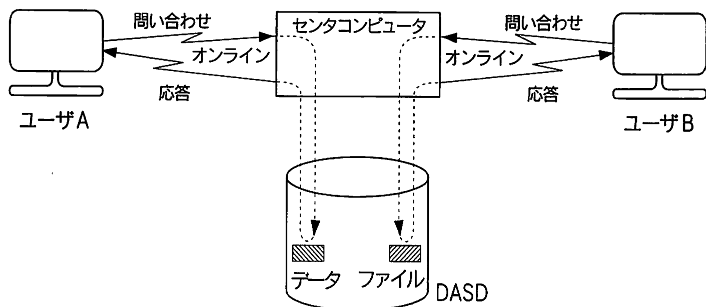
ファイル中のレコードの処理を，先頭から末尾に向かって順次に行う形式のファイル編成。順次編成ともいう。すべての外部記憶装置に作成可能であるが，典型は磁気テープ上のファイルである。通常，複数個のレコードをまとめたブロックとして扱う。記憶効率がよく，レコード全部一括して順次処理するのに適している。しかし特定レコードへの直接アクセス，書きかえ，レコードの追加などには難点がある。→(参)ファイル編成

順編成ファイル(sequential file)

記憶媒体上のデータの物理的な順序と，プログラムによるレコードの処理順序(論理的な順序)が同じ並びのファイル。順次編成ファイル，順ファイルともいう。磁気ディスク，磁気テープ，フロッピーディスクなど，ほとんどの入出力媒体で扱うことが可能である。またレコードの全件を扱う場合，アクセス時間も短く，媒体に高密度にデータを格納することができる。→(参)順編成，(対)直接アクセスファイル

照会(inquiry／enquiry)

問合せのこと。照会業務とは，オンラインシステムで端末からデータベースを照会し，その内容や更新結果を照会した端末に送り返す業務をいう。例えば，「在庫があるか」「座席があるか」との端末から問合せに，「有・無」を応答したり，予約するような業務。



障害管理(recovery management)

ハードウェアの故障やソフトウェアのバグなどによって正常な処理が困難となった場合，その影響を取り除きシステム全体，または一部の機能を正常な状態へもどすこと。障害回復，エラーリカバリともいう。次の4項

目の技術が要求される。

(1)障害の発生状況を監視し，状況を把握する障害監視技術
(2)障害に関する情報から，障害の原因をつきとめる障害究明技術
(3)的確な復旧作業の手順を決め，実施する障害回復技術
(4)事後のため，障害の記録，原因分析，保守計画を行う障害防止技術

コンピュータシステムの機能を回復するために，障害の発生を検出する監視プログラム，障害の発生した機能単位を切り離し処理の続行を可能にするための再構成，バックアップファイルによって誤りの生じる前の状態へもどす方法などがある。データファイルの回復にはアーカイブファイル(定期的な記録)，ジャーナルファイル(個々のアクセスごとの記録)，チェックポイントファイル(主記憶装置で展開されている内容の定められた時点での記録)等を利用する。

消去(erase)

①記憶媒体中のデータ領域を空データを表現するコードで置き換えること。消去した領域には，別のデータを記憶させることができる。EP-ROMでは強い紫外線にさらすことで記憶を消去する。磁気テープやフロッピーディスクなどでは，スペースコードに置き換える場合と，バルクイレーサという機械で完全に磁気記録を無くしてしまう方法とがある。

②文字や行を消す操作。削除に同義。→(同)削除

条件コード(condition code)

演算や比較などの命令を実行したとき，その結果(2数の大小関係，真偽など)を表すコード。演算命令では演算結果により正，ゼロ，負を示すコードが，比較命令ではオペランドの大小を示すコードをセットする。このほか，オーバフローや桁上げがあったときセットするコードもある。

条件つき飛越し(conditional jump)

プログラム中に指定された条件により飛び越し先を変えて，他の部分へ制御を移すこと。この指定を行う文，命令を条件分岐命令，条件付き飛越し命令(conditional jump instruction)，条件判断命令(decision)ともいう。

照合(collate／collating)

一連のデータのキー項目の大小関係を判断して処理を行い，記録の二つ以上の集合からひとつ以上の集合を作り出すこと。照合に関しての標準的な処理は，順番検査，突合せ，選別などである。→(関)順番検査，突合せ

詳細設計(detail design)

概要設計をもとに行う入出力帳票や処理手順の具体的な設計。例えば入力する原始伝票の内容や，プログラム仕様書の作成などがある。

→(同)内部設計

昇順(ascending order)

ソート処理で，配列やテーブルなどの特定キー項目について，小さい順番(英字ならABC順，カナや漢字ならJISコード順)にすること。正順ともいう。表計算ソフトでは，特に指定しなければ，昇順がデフォルト設定されている。大きさをもつ「数値」と文字列としての「数字」では順序が異なることに注意。

・数値の場合　12<21<35<134<226

・数字の場合　12<134<21<226<35

→(対)降順，(参)ソート

仕様書(specification／spec.)

ハードウェアやソフトウェアの仕様，性能などを文書化したもの。仕様書は，対象となる課題や記述法によってさまざまな書き方があるが，次の用件を備える必要がある。

(1)仕様の正当性，完全性，整合性などを備えていること

(2)必要かつ十分な情報を盛り込んでいること

(3)記述のしかたが簡潔でわかりやすいこと

(4)修正や変更がしやすいこと

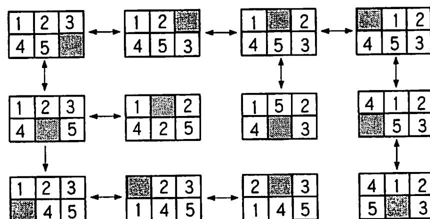
スペックは略称。一般的にはそのものが示す能力のことを表し，カタログなどに記載してある性能データをさすことが多い。→(類)スペック

状態遷移(state transition)

ある状態が他の状態と区別され，また時間の経過(または処理の結果)によって，別の状態に移ること。例えば，ゲームの展開，プログラム実行中の変数の値，ハードウェアの条件コードなどである。また，状態遷移図は状態間の移動を図式化したものである。

状態遷移図(state transition diagram)

情報の時間的順序を規定する図式のひとつ。現在の状態でどのような情報を受けたら，次にどのような情報を送り，どの状態に移るかを有向グラフ化した図。状態をグラフの頂点に，変化の方向を枝(有向，可逆なら双方向または無向)で表す。初期状態と終了状態は必ず存在する。もともとコンピュータの順序回路の設計に利用していたが，汎用性があるためプログラム言語の文法，オペレーティングシステムのタスク(プロセス)制御，データ伝送制御，通信プロセス間の情報の流れなどを正確に説明する手段として広く使用している。この他に，状態の遷移を決定表の形式にした状態遷移表も使用する。



シ

冗長検査(redundancy check)

送受信間で正しくデータをやりとりするために，本来のデータに別のデータ(冗長符号)を付加して誤り検出を行う方法。代表的な検査方法に，奇偶検査，群計数チェック，巡回冗長検査(CRC)などがある。

冗長検査符号(redundancy code)

冗長検査を行うために情報本体に付加した符号。もとの情報の長さ，検査の強度によって，冗長検査符号の長さは，1ビットの場合，1字の場合，1ワードの場合などがあり，パリティビット，チェックサム，検査文字，検査数字，チェックシーケンスなどと呼ばれる。なお誤りを検出したうえで訂正もできる冗長検査符号を特に誤り訂正符号という。

情報(information)

データを集め，比較したり整理したりして，その中から見つけ出したデータに含まれている意味や考察のこと。しばしば，データと情報とを区別なく使用する。データに何らかの処理を行って，意味あるもの，解釈できるものとして得られたものを情報と呼ぶことが多い。

情報科学(information science／computer science)

情報の性質，構造，論理，およびその処理方法について科学的な観点から研究する学問。特に情報をコンピュータで処理する際の理論を研究する学問をさすこともある。計算機科学との重なりも大きい。

情報検索(information retrieval：IR)

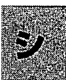
①大量の情報を集めて，その中から必要な特定の情報を体系的に整理，分類，保存，探索し，また，必要に応じて特定のデータを迅速，かつ正確に再編成，あるいは表示する技術とその過程のこと。

②研究や設計，または経営に必要な多くの資料，出版物，テストデータ，設計図などの整理と，これらのファイルから必要な情報をできるだけ早く取り出すこと。→(参)データベース，データバンク，電子ファイル

情報サービス産業(information service industry)

コンピュータユーザ企業のEDP部門に代わり，または援助する形で，ソ

フトウェア開発，プログラム作成，運用管理，受託計算，，要員教育，入力作業などを行う事業，企業，産業。コンピュータ利用の増加，ソフトウェア要求の増大とともに日本でも発達してきた。いわば情報処理技術を売物にする産業である。



情報処理(information processing)

与えられたデータや情報から目的とする情報を得るために行う，収集，処理，加工，整理などの一連の作業。データ処理と同義であるが，どちらかといえば情報を必要とする人間サイドからみた用語である。

情報処理技術者(information processing technician)

一般にはコンピュータシステムやソフトウェアの開発，運用，保守に関わる専門技術者。その職能は多岐にわたり，さらに細分化してシステムアナリスト，システムエンジニア，システムアドミニストレータなど数多くの呼称がある。

情報スーパーハイウェイ構想(information super-highway planning／National Information Infrastructure：NII)

アメリカのクリントン大統領とゴア副大統領が1993年2月に発表した，2000年までに米国全土の学校，図書館，病院，企業，家庭などをギガbpsクラスの高速光通信網でつなぐという構想。

情報スーパーハイウェイ構想が実現すると，マルチメディア情報をリアルタイムに送受信できるようになる。米国は，1950年代のハイウェイ道路建設で，巨大な自動車産業の創出，自動車通勤の普及や郊外住宅地の発展などの経済効果をあげ，経済を活性化した実績を有している。情報スーパーハイウェイ構想も，米国産業の国際競争力強化の切り札にすることが狙いである。情報ハイウェイを作ることにより経済の情報化を進め，新たな情報産業やサービス産業などの市場や雇用の創出をし，生活水準の向上，在宅勤務や電子ショッピング，電子新聞などの実現，さらに教育や医療などの社会サービスの充実などを計画している。さらにNIIから地球規模のGIIへと構想を拡大している。　日本でも，この構想が発表されたことを受けて，1994年8月内閣に「高度情報通信社会推進本部」を設置し，1995年2月に「高度情報通信社会に向けた基本方針」を発表している。

情報リテラシ(information literacy／computer literacy)

コンピュータシステム，ネットワークシステムで情報を扱う基本的な知識と技能，および情報利用に対する倫理観。情報社会では，多くのユーザがコンピュータを使用しており，専門家だけでなく，一般の人々にもコン

ピュータの正しい知識と使いこなす技能と情報利用に対する倫理観が必要となる。リテラシとはもともと文字の読み書きの能力の意味で，コンピュータの操作等も情報授受の基本的な能力として必要なものになったとの認識ができつつある。



証明機関(certification authority：CA)

暗号化のための公開かぎ証明書の発行，管理，配布を運用する機関。認証局ともいう。なりすまし犯罪を防止するために，ネットワーク上に公開されているかぎが本当に当人，当事者のものかを証明する。正規の利用者Aは自分の情報と公開鍵を証明機関に登録しておき，証明機関の秘密鍵によるディジタル署名つきの証明書をもらう。

正規の利用者Aは自分の情報と公開鍵を証明機関に登録しておき，証明機関の秘密鍵によるデジタル署名つきの証明書をもらう。Aに通信文を送る利用者Bは，証明機関の公開鍵で利用者Aの情報と公開鍵を得ることができる。A以外の者がAを称して(Aになりすまして)別の公開かぎをネットワーク上に公開しても，その鍵は証明機関に登録されている正規の鍵ではないことを識別できる。

商用データベース(commercial database)

営利目的で情報を第三者に提供するデータベースサービス。現在では電話回線によるパソコン通信でデータベースにアクセスするオンライン方式が普通である。サービスの内容は，政治経済や科学技術などの情報を蓄積した専門情報データサービスと，一般のニュースや電子メール，電子掲示板などのサービスを中心としたコンシューマデータベースの2種類に分類できる。さらに専門情報データサービスは検索結果が利用者が求める情報そのものであるファクトデータベースと文献の抄録のみのリファレンスデータベースに分けることができる。

初期化(①initialize／clear，②media initialize)

①プログラムやサブルーチンの実行に先だって，カウンタやスイッチその他のデータ領域をある値に設定すること。文字属性の領域には空白を，数値属性の領域にはゼロを設定する。→(類)クリアする，(関)初期値

②ハードディスク，フロッピーディスクや磁気テープ等の記憶媒体を(すでにデータを記録してある時はその内容を消し)新しいファイル用媒体として使用可能な状態にすること。この処理はフォーマットと呼ぶことが多い。→フォーマット

初期設定(customize／preference)

新規のハード，ソフトを使用開始するとき，規定の動作をするように機

能の設定をすること。メーカが多数のユーザの使用を想定して，出荷時の製品に設定した標準的な設定をエンドユーザシステムに合わせること。システム設定，カスタマイズともいう。

(1)ハードウェアについては，例えばOSやデバイスドライバをインストールし，起動時や運用時の状態を指定すること。

(2)ソフトウェアについては，例えばワープロソフトの起動時における入力モード，標準出力文書の体裁などの設定。ページの字数，行数などの情報である。→(類)システム設定，カスタマイズ

初期値(initial parameter／initial value)

①繰返しループを実行するに先だって，制御変数，領域に設定する定数，条件の値のひとつ。制御変数は，この値をもって繰返しループに入り，終値を越えると繰返し処理を終了する。初期値を入れることを初期値の設定という。

②COBOLでは，プログラムの実行にあたって，最初にデータ項目に与えられる値。初期値は，VALUE句やINITIALIZE文などによって与える。初期値を設定していない項目の値は，そのときの主記憶装置の状態によって異なるため，何になるかは特定できない。

書式(format)

①データ媒体に入出力する際のデータの形式。例えば，磁気テープや磁気ディスク等の記憶媒体上のデータの並び，ディスプレイ画面のレイアウト，出力帳票やワードプロセッサの印刷文書の体裁(見出し，罫線，用紙サイズ，文字の大きさ，金額表示のルール)等である。

②文書処理では文書の体裁そのもの，または体裁を定める情報。例えば用字の書体やサイズ，縦書き，横書きの区別，1ページの字数，行数などの情報である。

書式制御(format control)

印字，表示，記録するデータの形式や位置を指定するための処理。

①出力命令とともに用いる書式制御文(例えばFORTRANのFORMAT文など)の内容。

②プログラミングで，出力するページ数や行数，桁数の指定やデータの並びの指定。

③文書処理の書式制御では，出力する文字の前後に書式を指定する書式制御文字を挿入する手法と，出力する文字列とは別に書式制御情報をもつファイルを準備し，出力機器を制御する手法がある。

書式設計(format design)

データ媒体に入出力する際のデータの形式を設計すること。例えば，ディスプレイ画面のレイアウト，出力帳票や印刷文書の体裁(見出し，罫線，用紙サイズ，文字の大きさなど)等をシステムの目的に合わせて決定する。

ジョブ(job)

ユーザ，オペレータがコンピュータに処理させるひとまとまりの仕事(作業)の単位。複数個の関連をもつジョブステップからなる。



ジョブ管理(job management／job control)

ジョブは，ユーザ，オペレータがコンピュータに実行させる仕事の単位。ジョブ管理プログラムは，受けた複数のジョブの実行の準備や後始末，ジョブを読み込んで実行に必要な資源の割当て(アサイン)やスケジューリングを行うなど，オペレータとコンピュータとの間の情報交換を担当する。

処理(process／computing)

①データ，データ群に演算を施すこと，関係を作ること。入力したデータに対し，あるいはデータ間であらかじめ定めた規則で加工(例えば演算，ソート，マージなど)すること。ソフトウェアの開発では，プログラムのアセンブル，コンパイル，編集，ファイルの更新などの操作を広く含む。

③データ処理の工程。複数の工程では，中核となる処理を主処理，主処理以前に行う処理を前処理，主処理以後に行う処理を後処理という。

④データ処理システムの特徴的な形態や方式。集中データ処理，分散データ処理，オンラインリアルタイム処理などという。

処理・加工セグメント(processing and manipulating segment)

制御セグメントから呼び出され，引き渡されたデータを処理，加工するセグメント。モジュールのセグメント化を進めるとき，処理条件を明確にした制御セグメントで，実行条件の整理や実行順序の制御を行い，従属モジュールでデータの処理や加工を行う。→(対)制御セグメント

処理記号(process symbol)

(1)流れ図記号のひとつで，基本処理記号と個別処理記号がある。処理記号：任意の処理を表す長方形の記号。

(2)個別処理記号：処理の内容によって書き分ける処理記号で，定義済み処理，手作業，準備，判断，並列処理，ループ端記号がある。

処理スケジュール(operation scheduling)

データ処理のタイミングを考えて，定期的な処理も含め，多くの仕事の順序，処理時期，区切りなどの日程や時間をスケジュールすること。

JCL，ジョブネットの作成，バックアップファイルの作成なども含む。

資料調査(documents research)

現状分析の際，本格的な調査，実態調査に先だって行う文書や文献の調査。システム化の対象となる業務に直接的に関係する文献，法規や規定，マニュアルや手順書，商品カタログ，取引先名簿，各種伝票などの文書，さらには業界の統計資料，競合他社の状況などを収集，調査すること。



シリンダあふれ域(cylinder overflow area)

索引順次編成ファイルのオーバフロー領域(あふれ域)のうち，各シリンダごとのシリンダあふれ域。つまり，そのシリンダ内のトラックからあふれたレコードを収容する。なお，特にシリンダあふれ域を設けず，独立あふれ域のみを設けて，あふれた全レコードをそこに収容する場合もある。

シリンダ索引(cylinder index);

各シリンダ内のレコードの最大キーとシリンダ番号が記録された索引。ひとつのファイルにひとつあり，検索対象のレコードがどのシリンダに存在するかを特定するために用いられる。

人工言語(artificial language)

その言語を使用するための規則(文法)を厳密に定めた言語。自然言語に対し，プログラム言語(COBOL，FORTRAN，C，PL/I等)，ロボット制御言語やエスペラント語のように，人工的に作られた言語をいう。最近では，命令に対応して書き方を決めた日本語を使用して作成したプログラムを，通常のプログラム言語によるプログラムに変換するシステムの開発もあり，より自然言語に近い形でプログラムを作ることができるようになっている。→(対)自然言語

人工知能(artificial intelligence：AI)

人間のような学習能力や推論の機能をもつ知的なコンピュータシステム。定型化できない問題解決をするプログラム群をさすこともある。専門家の知識をもとに作成した知識データベースに基づいたエキスパートシステムも実用化の段階に入りつつある。人工知能システムには従来からのノイマン型コンピュータではなく，第5世代コンピュータの開発理念ともなっている非ノイマン型コンピュータも有望。また知識データベースを取り扱い，推論の仕方を定義する技術者(ノレッジエンジニア)も必要。

人事システム(personal management system)

企業の人的資源の調達，開発，維持等における管理活動を実行，または支援するコンピュータシステム。その内容は多岐にわたり，次のようなシステム例がある。

(1)給与計算，課税，保険，年金などの人事ルーチンの自動化システム。
(2)人事統計(部門別の賃金，出勤率，事故率など)の集計システム
(3)人事情報(学歴，職歴，資格，教育訓練歴，特殊技能，考課など)の管理システム
(4)福利厚生施設(寮や社宅，体育館など)，安全衛生(健康診断など)，モチベーション(配転希望や提案，申告など)に関する情報システム

シンタックス(syntax)

プログラム言語の文の構造を支配する規則。特にソースプログラムの文を正しく構成するための規則で，プログラム言語を記号の列とみなしたとき，正しいプログラムはどのような記号列であるべきかを規定する。

シンタックスエラー(syntax error)

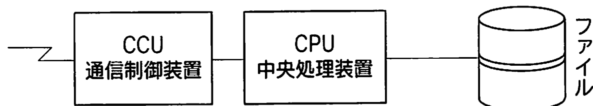
ソースプログラムをオブジェクトプログラムに変換する過程で，コンパイラやインタプリタが見つけた文法上のエラー。予約語の綴りミス，オペランドの過不足，構文の矛盾など。コンパイラは通常，エラーの発生する箇所や種類を指摘，出力する。これをエラーメッセージという。シンタックスエラーがなくなればオブジェクトプログラムを生成できるが，論理エラーは，テストデータを与えてプログラムを実行してみないと発見できない。→(参)論理エラー

進捗管理(progress management／expediting／follow up)

計画した工程，文書，納期，品質，コストなどと，実際の進捗状況の隔たりをなくすこと。予定ー実績の管理(予実管理)，差異分析を行い，問題があれば対応策をとる。問題がある場合，その原因は，計画，チーム運営，技術水準，要員数，作業時間など多岐にわたる。

シンプレックス構成システム(simplex system)

1台のCPU(中央処理装置)に，最小の構成でファイル装置や通信制御装置を接続したシステム。単一システムともいう。コストは安いが，どれかの機器の障害時には全面停止になるという欠点がある。システムの信頼性は低い。



シンプレックス法(simplex method)

線形計画問題の解法のひとつ。単体法ともいう。ひとつの基底解から出発し，単体判定基準を用いて適宜基底変数の入替えとそれにともなうピボット演算を行って，目的関数の値を最大(または最小)にする最適解を求め

る方法。これは一種の山登り法であって，ひとつの頂点から出発して目的関数の値の増加(または減少)する方向でその近傍の頂点に移り，もはやこのような移動ができなくなったところで停止するというやり方である。1947年にダンチッヒ(G.B.Danzig)によって開発された。

シンプレックス表(simplex tableau)は，シンプレックス法を解くための表。単体表ともいう。制約条件(スラック変数，人為変数を含む)の中の各変数の係数と制約値，目的関数(スラック変数，人為変数を含む)における各変数の係数が記入されている。そして，山登り法により簡単に最適解を得るためのもの。

信頼性・信頼度(reliability)

コンピュータシステムや機器，部品などの機能単位が一定条件のもとで一定期間，安定に実行する能力。コンピュータシステムの信頼性は，ハードウェア，ソフトウェアに依存し，RASの概念を生み出した。信頼性向上のために，信頼性設計，誤り制御方式，障害対策，予防保全など多方面でのくふうを積み重ねている。

信頼性評価尺度(reliability measurements)

システムの信頼性がどの程度であるかを評価する尺度。個別の機器については，N個の正常なサンプルが規定時間内にr個故障するとき，信頼度は，(N−r)／N で表す。システムは複合体であり，長期的に見てシステムが稼働可能な状態にある時間の割合で可用性(availability)を目安にすることもある。次にあげるものが代表的な尺度である。

(1)平均故障間隔(mean time between failures : MTBF)

システムが正常に動作し続ける(ある故障時から次の故障時までの)時間の平均。MTTRと合わせて稼働率や不稼働率の計算に使用する。

(2)平均修理時間(mean time to repair : MTTR)

故障が発生した場合に，その修理に要求する平均時間。システムの故障が継続する時間の平均。平均ダウンタイム(mean down time)ともいう。これが大きいことは，システムの信頼性が低いことを意味する。

(3)稼働率(可用性)(availability)

システムが正常に動作している時間の割合。稼働率が高いほど信頼性が高いことを表す。オンラインシステムが大規模化してくると，信頼性に対する要求が厳しくなってくる。システム設計時，各構成要素の稼働率と構成要素の接続形態から，システム全体の稼働率を予測しておかなければならない。なお，個別構成要素の稼働率は次の式で求める。

稼働率＝ MTBF／(MTBF＋MTTR)

(4)不稼働率(unavailability)

システムが正常に動作していない時間の割合または確率。次の式で表す。

不稼働率 ＝ MTTR／(MTBF＋MTTR)

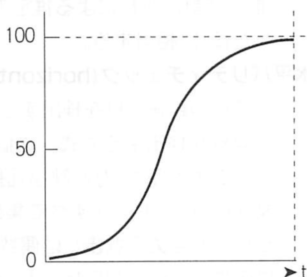
(5)信頼度関数…ある時刻tにおいてシステムが正常である確率。この関数を使えば，通信衛星ようなシステム(非修理系)の平均寿命が次の式でわかる。

$$\text{修理できないシステムの平均寿命} = \int_0^{\infty} R(t)dt$$

R(t)は信頼度関数

信頼度成長曲線(software reliability growth curve)

テスト終了時を判断する目安のひとつで，テストを繰り返すたびに発見したエラー件数をグラフ化したもの。テストと修正によりシステムの信頼度が高まって行くようすは，ある時点までに発見されたエラー件数の累積に対して，100%に近付いて行く曲線の傾きである程度把握できる。曲線の傾きが水平線に近づく度合で信頼度を判定する。



ス

垂直パリティチェック(vertical parity check)

符号の転送誤りを検出する冗長検査方式の一種で，一文字単位でデータ誤りを検出する方式。送信時には一文字について，パリティビットと呼ぶ冗長符号 1ビットを付加する。→(参)奇偶検査

垂直分散構成(vertical distributed processing system)

分散データ処理を行うシステムの構成法のひとつで，ホストコンピュータに接続した複数のサブホストコンピュータ，さらにその下位に接続した多数の端末の間で，業務を分散して処理する方式。例えば，地域に分散する多くの事業所，事業所をまとめる地方支店，それを統括する本店で構成する業種，業務では，各部署の業務は連携していると同時に，独自の処理が必要である。この場合，業務の構成とコンピュータの構成を重ねること

で分散処理が可能であるが，その構成は，組織と同様の垂直的な構成になる。→(対)水平分散構成

スイッチ(program switch)

プログラムの中で条件分岐の目印として利用する変数の名称。スイッチは分岐を制御し，分岐点に到達する前に値を設定しておく。フラグ(flag)，スイッチ点(switch point)，プログラムスイッチともいう。

ス

推定(estimation)

総計量より，その標本(試料)が抜き取られた母集団の母数を推定すること。点推定と区間推定とがある。推定値(observed value)は，観測値から母集団に関して，規定の信頼水準で推定して求めた特性値。これは統計的推定(統計的推移による推定の方法)の区間推定値(ある範囲のもとに推定された値)に相当する。

水平パリティチェック(horizontal parity check)

符号の転送誤りを検出する冗長検査方式の一種で，データブロック単位に誤りを検出する方式。送信側でデータブロックごとにBCC(ブロックチェックキャラクタ)と呼ぶ冗長符号を付加する。BCCは，ブロック内の各文字のnビット目をすべて集め，その中で1であるビット数がBCCの対応ビットも含め奇数または偶数になるようなビット列から構成する。受信側は各文字とBCC対応ビットの1であるビット数を調べて，送信にエラーがなかったかチェックする。水平パリティチェックも1ビットの誤りしかチェックできないが，垂直パリティチェックを組み合わせることで，2ビット以上の誤り検出ができる確率が高くなる。→(参)奇偶検査

水平分散構成(horizontal distributed processing system)

ホストコンピュータを中心とした階層型(垂直型)の分散処理システムではなく，同規模のコンピュータを多数並べ，それぞれ得意とする機能を分担して処理する分散処理方式。コンピュータ間に基本的には上下関係はなく，それぞれ対等であるため水平型と呼ばれる。水平分散処理システムの代表的なモデルにクライアントサーバモデルがある。→(対)垂直分散構成

数字検査(numeric check);

→(同)ニューメリックチェック

数値制御(numerical control：NC)

→(同)NC制御

スキーマ(schema)

データベースに蓄積するデータそのものではなく，データの意味や操作のルールの定義(メタデータの集まり)をいう。すなわち，データベースの

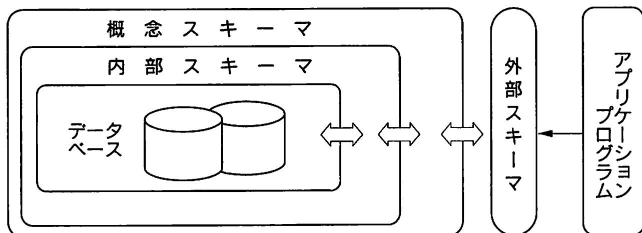
論理構造，格納構造，物理構造の定義と記述。

代表的なスキーマには，CODASYLのDBTGが提唱した2層スキーマと，ANSI／X3／SPARCが提唱した3層スキーマがある。

(1)概念スキーマ：コンピュータのことや分野などのことから離れて，データそのもののデータベース全体の論理構造を記述したもの。

(2)外部スキーマ：データベースを操作する応用プログラムごとに，データの格納構造を記述したもの。

(3)内部スキーマ：概念スキーマと外部スキーマを，コンピュータ上で実現するための物理構造を記述したもの。



スクラッチパドファイル(scratch pad file)

ジョブステップ間のデータの受渡し，主記憶の記憶容量不足のカバー，暫定のコピーなどのために一時的に作られるファイル。

スクリーンエディタ(screen editor)

カーソル移動キーとマウス，キーボードなどを使用して，画面上にプログラムや文書を編集(作成，更新，削除)し，レコードやファイルを作成するプログラム。→(類)テキストエディタ，(参)エディタ

スクリプト(script)

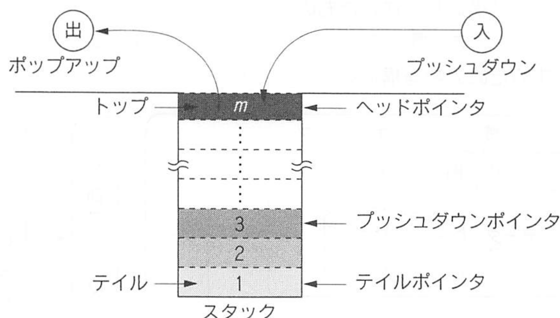
定型化された手順を記述した命令群。プログラム言語によるプログラムとは異なり，マクロ言語やスクリプト言語のようなコマンドで表現する命令。定型的処理の自動運転を行うMD-DOSのバッチプログラムやUNIX系OSのスクリプトファイル，マクロ機能のあるアプリケーションのマクロ命令の記述などをさす。

スタック(stack)

後入れ先出し(LIFO)のデータ構造。後入れ先出し記憶装置または記憶領域をさすこともある。複数のデータを順次に記憶するが，取り出せるデータは常に1件で，最も後から記憶したデータのみである。列の一端だけで追加や取り出しが可能である。例えば，サブルーチンの戻り番地の管理，

多重の割込みを処理する入れ子の構造のデータを処理するために，後入れ先出しの記憶装置(スタック)が必要になる。

新しく記憶するデータを古いデータに積み重ねることをプッシュダウン(push down)，または単にプッシュ(押し込み)という。逆にスタックから最も新しいデータを取り出すことをポップアップ(pop up)，または単にポップ(跳ね上げ)という。→(関)リスト処理，LIFO

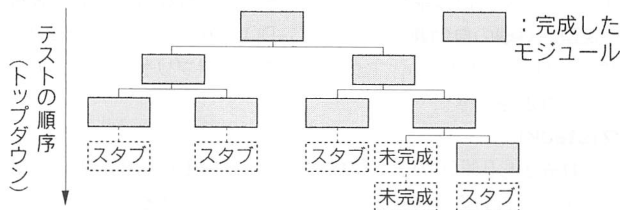


スタックポインタ(stack pointer：SP)

配列でスタックを実現するとき，アクセスする場所を指定するポインタ。記憶場所を示すアドレス，または添字であり，特に記憶構造がスタックの場合，こう呼ぶ。スタックを配列で実現するとき，スタックポインタになるアクセスアドレスを特にヘッドポインタ，スタックの末尾のアドレスをテイルポインタという。

スタブ(stub)

上位レベルのモジュールまたはモジュールの集まりをテストするために，下位レベルの未完成モジュールのインタフェース部分だけを設定したテスト用のプログラム。→(関)トップダウンテスト



スタンドアロン(stand alone)

パソコンを他のコンピュータやパソコンと接続しないで，単体で利用する形態，または独立型の機器。ソフトウェアについても，他のプログラムと連係しないで単体で使用する場合，スタンドアロンという場合がある。

ステートメント(statement)

→(同)文

ストア(store)

①レコードをファイルに書き込むことなど，データの記憶装置への格納。反対に記憶装置からデータを取り出すことをロードという。→(同)アンロード，(対)ロード

②アセンブラ命令のストアは，レジスタ内容をメモリへ格納すること。

ストアドプロシジャ(stored procedure)

データベースサーバへのアクセス回数を低減するための一手法。典型的な一連のSQL文をあらかじめコンパイルしておき，サーバのデータベースに格納しておく。そして，クライアントは必要なコンパイル済みのSQLを呼び出し実行する。これにより，SQL文をそのまま実行するより通信量，通信回数，クライアントとサーバの負荷とも大幅に少なくすることができる。

ストラクチャードコーディング(structured coding)

構造化定理に基づいたコーディングのこと。3種の基本制御構造(順次，選択，繰返し)を基本にして論理を組み立て，GOTO命令を極力使用しないようにすることにより，分かりやすいプログラムを作成する。

ストリーマ(streamer)

大容量外部記憶装置の一種。磁気テープが記憶媒体で，数ギガバイトまで記録できる。おもな用途は，ハードディスクのバックアップ，アーカイブファイルの作成など。従来の磁気テープ記憶装置との大きな違いは，テープがカートリッジ等に収納したまま記憶装置へ出し入れでき，テープへのアクセス速度が速くなった点にある。

ストリング(string)

→(類)列，文字列，コード

スパイラルモデル(spiral model)

ウォータフォールモデルとプロトタイプモデルの両方の手法を折衷したモデル。開発の各期階で必要とする技術者の技能が異なるため，ウォータフォールモデルでは初期と後期にシステムアナリストやシステムエンジニアが不足し，中期にプログラマが不足する現象が起きるが，これを解消するため，開発の初期から独立性の高いモジュールについては設計，プログ

ラミング，テストを行い，順次にシステムを開発する手法。

スパンドレコード(spanned record)

ファイルを構成しているレコードのうち，隣接する複数のブロックにまたがっているレコード。スパンレコードともいう。

ス

スペーシングチャート(spacing chart)

コンピュータで処理した結果をディスプレイ画面や用紙に出力する場合に，どの位置にどのデータを印字するかなど，形式を設計するために用いる用紙。1画面，1ページの文字数，行数に合わせた罫線が印刷してある。→(関)出力設計，帳票設計，画面設計，画面レイアウト

スペック(specification／spec.)

specification(仕様)の略語。本来はハードウェア・ソフトウェアの仕様書のこと。しかし，一般的にはそのものが示す能力のことを表し，カタログなどに記載してある性能データをさすことが多い。→(類)仕様書

スループット(throughput)

①単位時間当たり処理できる仕事量。高価なコンピュータ資源を効率よく使用するため，オペレーティングシステムでは，スループットの向上が主要な目標になる。この実現のため，マルチプログラミング技法，スプール技法などがある。

②データ転送の効率(単位時間当たりのデータ通信量)の意味でも，用いられているようになっている。

スレッド(thread)

Windows，UNIXなどマルチタスク処理を行うオペレーティングシステムでの処理の基本単位。プロセス(タスクとほぼ同じ意味)を細分化したもので，スケジューリングを行う単位。

スレッドはCPU時間を割り当てる最小単位である。マルチタスクの際にはプロセスよりも高速に生成でき，より小さいためスワップの負荷が軽減でき，マルチタスク処理の実行効率を上げることができる。プロセスをスレッド単位に分割できるOSをマルチスレッドOSと呼んでいる。

正規化(normalization)

①ある量を定められた範囲内に収まるよう調整すること。正規化した形を正規形という。

②ある数の浮動小数点表示で小数部があらかじめ定められた範囲内に入るように指数部と小数部を調整すること。例えば，10進数で小数部xの範囲を $1 \gg x \geqq 0.1$ (xは1以下の最大値)に定めてある計算機で演算結果として 0.00632×10^8 を得たときは，これを 0.632×10^6 に変更する。

③関係データベースで，データの独立性を保つために行うデータのグルーピングのこと。関係データベースでは，重複データの排除，データの更新，挿入，削除の影響の排除などデータの独立性の維持が重要であり，この条件を満たすために次のようなグルーピングを行う。

(1)第1正規化：繰り返しのある項目の独立化する

(2)第2正規化：複数のキー項目が揃ってから初めて意味がある項目以外を独立化する

(3)第3正規化：他の項目からの間接演算で同じ内容が得られる項目を除去する

正規形(normal form)

関係データベースで正規化をしたデータ。

(1)第1正規形(first normal form)

データベースを構成するレコードが，それ以上分解できないところまで分けられているとき，その表は第1正規形という。すなわち，繰り返しのある項目がなく，すべて独立した項目の表現になる。

(2)第2正規形(second normal form)

あるデータ項目(a)が決まると，他のデータ項目(b)が一意に決まるとき，bはaに機能従属するという。このような機能従属がないとき，第2正規形の表という。すなわち，必要最小限のデータの組合せになるように表を分割することである。

(3)第3正規形(third normal form)

あるデータ項目(a)が決まると，他のデータ項目(w)が決まり，項目(w)が決まると，他のデータ項目(b)が決まるとき，bはaに推移従属するという。表に推移従属がないとき，第3正規形という。すなわち，食い違いが生じないように表を分割することである。

正規表現(regular expression)

UNIX系OSのエディタやシェルで使う文字列の抽象表現。例えば，ピリオド(.)で任意の1文字，アスタリスク(＊)で文字の繰返し，ダラー($)で行末を示す。文字列の検索や置換に便利な機能である。

例えば，ｊ．＊ｇ＄は「ｊで始まりｇで終わる文字列で，かつ行末にある文字列」を意味する。正規表現に使用する文字をメタ文字といい，ワープロやエディタの機能に取り込んでいるソフトも多い。MS-DOSでは，ファイル名と拡張子の指定に2種類のワイルドカード，疑問符(？)が任意の1文字，アスタリスク(＊)が任意長さの文字列を使用できる。

正規分布(normal distribution)

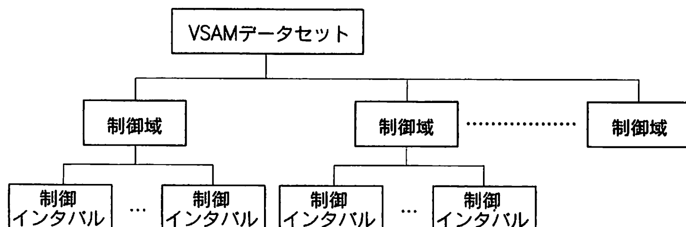
確率密度関数が，

$$f(x)=\frac{1}{\sqrt{2\pi}\,\sigma}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

で表される連続分布。正規分布は，現実の世界で起こるさまざまな事象が近似されるので，統計ではしばしば利用される。

制御インタバル(control interval：CI)

VSAMデータセットを構成する論理的な構成要素のひとつであり，制御エリア(CA)を細分化した領域。制御インタバル(CI)がいくつか集まって制御エリアを構成する。ひとつのCIには，複数の論理レコードが記憶され，CIはVSAMにおける入出力操作の単位となる。ひとつのデータセットを構成する制御インタバルは固定長であるが，これを構成する論理レコードは固定長の場合も可変長の場合もある。制御インタバルには，レコード定義フィールド(record define field：RDF)と制御インタバル定義フィールド(control interval define field：CIDF)がある。RDFは論理レコードを記録するフィールドであり，CIDFは空きスペースの位置と大きさおよび各論理レコードの長さと制御情報を記録するフィールドである。各フィールドは固定された位置にあるのではなく，制御インタバルの前と後から作成されていく。



制御エリア(control area：CA)

VSAMデータセットを構成する論理的な区域。レコードを記憶する。複数の制御インタバルによって構成される。また，複数のCAが集まってひとつのデータセットを構成する。

将来のレコード追加に備え，VSAMデータセットの終わりには未使用の制御エリアを用意する。これを空き制御エリア(空きCA)といい，空きCAが無くなった場合は，新しいCAを設定することができる。

セ

制御構造(control structure)

→(同)基本制御構造

制御セグメント(control segment)

ひとつのモジュールの中でも，いくつかの機能部分に分けて考えるとき，細分化した単位をセグメントという。上位のセグメントは，モジュール全体の管理を行うだけの場合，特に制御セグメントという。制御セグメントは従属セグメントを呼び出すだけであり，実際のデータの処理・加工は従属セグメントで行う。

制御用マイクロコンピュータ(control micro computer)

プリンタやハードディスク装置などの周辺機器・家庭用電化製品に組み込まれた制御を専門とするマイクロコンピュータ。ROMに記憶されたプログラムとCPUでさまざまな機能を提供する。

生産管理システム(production management system)

生産管理は，生産工程と機械，設備，作業の管理を行うことである。たとえば，生産計画(原材料や資材の所要量，製造日程の決定，時間軸管理，在庫管理など)，生産制御(生産工程や生産機械の管理)，生産時点情報管理(生産活動の状況把握)などを行う。これらの諸活動を実施，または支援するコンピュータシステムを生産管理システムという。

生産自動化(production automation)

生産工程の自動的手段による作業・処理の遂行およびそのための変更(自動化)。次のようなものがある。

(1)計画，設計などデザインオートメーション

(2)装置工業のプロセスオートメーション

(3)機械工業のメカニカルオートメーション

(4)生産にともなう事務分野のビジネスオートメーション

生産-物流-流通システム

生産者と消費者間の物流を統合的に扱うシステム。物流EDIともいわれる。物流EDI(physical distribution EDI)は，物流分野に適用したEDIのこと。

受発注業務，在庫照会，入出荷データ，貨物追跡管理，請求管理などのデータを電子化し，コンピュータや通信回線を利用し処理すること。　物流EDIでは，荷主・倉庫・運送業者・金融業者など多数の事業者の相互間でのデータ交換が必要であり，標準化されたEDIプロトコルが使用されている。→(関)電子データ交換

セ

生産-流通統合システム

ネットワークを介して生産部門のシステムと流通部門のシステムをつなぎ，両者間の情報を交換しスムーズな生産と物流を目指す統合システム。

整数演算(integer arithmetic operation)

整数型(integer type)のデータ形式で行う演算。小数部は扱わない。10／3を3とするような小数部無視の演算で，浮動小数点演算などで問題になる桁落ちはないが，扱える数値の範囲は狭い。

整数化(round)

①実数型から整数型への型変換。小数点以下の値を切り捨てる。整数演算をする場合，FORTRANやCOBOLでは結果の小数点以下を切り捨てる。

②小数から整数にすること。一般的には小数点以下の四捨五入，切上げ，切捨てなどの方法がある。小数部を切り捨てる関数INT(　)を用いて整数化する場合通常は以下のような式を用いる。

・四捨五入…… INT(X+0.5)

・切上げ……… INT(X+0.99999…)

静的(static)

プログラムのコンパイル時に決定される性質のこと。識別子の外部結合や内部結合は静的な性質である。また，staticな記憶クラスの変数は，コンパイル時に割付けが終了している。

精度(precision)

①正確度と精密度をあわせて精度という。測定値の場合，真値との誤差が小さいほど正確度が高い(正確である，accurate)といい，測定値間のばらつきが小さい測定値ほど精密度が高い(精密である，precise)という。この正確度と精密度をあらわす度合いが精度である。→(関)誤差

②コンピュータの場合，実数型の数値に有効桁の精度がある。高い精度が必要な場合，倍精度，多倍精度などの指定をする。→(関)有効桁，単精度，倍精度，多倍精度，丸め誤差

③通信中の桁落ちの問題，画像処理の際の分解能(解像度)の問題など精度に関わる注意すべき点がいくつかある。主に信頼性という観点から精度が問題になる。

正当性(correctness)

アルゴリズム，またはそれを表現したプログラムが，与えられた仕様を満たしていること。次の二つのことが必要である。

(1)プログラムの実行過程のどの部分も，常に仕様の与えた条件を満たす

(2)プログラムの実行が有限回(有限の計算量)で終了する

プログラムの正当性を証明することを検証(verification)という。

セ

性能(performance)

コンピュータシステムの性能は，処理能力(throughput)とも呼ばれ，通常は単位時間当たりに処理できる仕事量で表す。ハード的には高速な処理装置，大容量の記憶装置などが，ソフト的にはオペレーティングシステムの機能，プログラムの効率性などが影響するため，性能の測定はさまざまな視野の下で行われる。

性能管理(performance analysis)

システムの性能は，ユーザやデータ量の増加などで低下することがあるので，稼働中の性能を監視，分析し，信頼性を保持すること。将来の処理量予測，性能向上のための現状評価や問題点の探索をも含む。

性能テスト(performance test)

開発したコンピュータシステムが所定の処理能力を発揮するか検査すること。システムにより，処理能力の評価基準は異なるが，通常は単位時間当たり処理できる仕事量，入力から出力を得るまでの応答時間，過大なデータを与えた場合の遅れなどが設計基準を満たすか，運用上の問題になるかを調べる。

西暦2000年問題

西暦年月日表現をyymmddのように，年部分を下2桁の数字で済ませたことが西暦2000年以後のコンピュータシステム運用に悪影響を及ぼすこと。クロック問題ともいう。

システムが，例えば数字02を1902年か2002年か判断できない場合に生ずる諸問題である。西暦年を4桁表現に変更するか，区別する入出力ルーチンを追加するなどの方策をとる。既に利用しているプログラム上の問題箇所を検索するツールなども販売されている。

整列(sort／sorting)

ファイル中のレコードを指定した項目について昇順あるいは降順に並べ変えること。整列の際に，補助記憶装置を使うか使わないかにより，外部分類と内部分類に分けられる。ソート，分類ともいう。

セ

セーブ(save)

一般的には特定の情報を破壊や消滅から守るために別の記憶媒体，別の場所に複写(または移動)すること。退避ともいう。特に主記憶装置内のプログラムやデータを磁気ディスクなどの外部記憶装置に記憶するときに用いる語。セーブの形式には，アスキーセーブとバイナリセーブがある。

(1)アスキーセーブ：アスキー形式ファイル(テキストファイル)でプログラムなどをフロッピーディスクに記憶すれば，ほかのパソコンでも読み取ることができる。

(2)バイナリセーブ：プログラムやデータを主記憶装置内部のデータ形式(2進コード)のままセーブすれば，ファイルが小さく，ロードの際にメモリ上に呼び出す時間も短くてすむ。

セキュリティ(security)

コンピュータ資源を破壊，障害，盗用，悪用などの犯罪等から守ること。コンピュータセキュリティ，安全性ともいう。主としてハードウェア機能の破壊(事故・災害による)，ソフトウェア機能の破壊(運用ミス，負荷集中等による)のほか，不正使用，機能侵害，データ破壊，改ざん等の悪用に対する防止対策をさす。

(1)アクセス者の正当性を確認する認証技術

(2)データそのものをアクセス者から隔離する技術

(3)アクセス者を特定できる監視技術

具体的には，ハードウェアを火災，出水，埃などから守る設備，IDカード設備の整備，パスワード，コピープロテクト，コールバック，暗号化，オペレーションログの実施などをセキュリティレベルに応じて行わなければならない。これに加え，システムダウンにそなえるバックアップ技術や回復技術も重要な安全性確立の技術である。

→(参)プライバシ保護，プロテクト

セキュリティ管理(security management)

コンピュータシステムのセキュリティを維持するために，管理基準や管理体制を作り，教育，訓練，監査などを行うこと。セキュリティを脅かす要因が悪意ある人間の場合，対策がいたちごっこになる可能性もある。

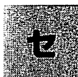
セクタ(sector)

磁気ディスク，ディスクパック，フロッピーディスクの各記憶媒体は，トラックを複数に分割し，任意の場所をアクセスできるようにしてある。その分割したトラック上の最小単位をセクタという。トラックは等分割する場合や任意のサイズに分割する場合がある。大きなデータは複数のセク

タにまたがって記録し，小さなデータは1セクタに記録する。しかし，すでに記録したデータがあると，他のデータをそのセクタに記録できない。

セグメント化(segmentation)

分割したプログラムモジュールの内部もさらに幾つかの機能部分に分けて考えることができる。このモジュールを細分化した部分をセグメント(segment)という。セグメントは，論理的にまとまりのあるステートメント(文)の集まりである。モジュールをセグメントに分割することをセグメント化という。



世代(generation)

①コンピュータの技術的な発展を，使用する素子，技術などにより区分した時代のこと。

(1)ハードウェア面では，論理回路素子が真空管，トランジスタ，IC，LSIと進歩してきたことに合わせてそれぞれ，第1世代(1950年代まで)，第2世代(1960年中頃まで)，第3世代(1970年頃まで)，第4世代(1970年以降)と分類している。現代は適用している技術の世代区分が難しく第4.5世代などということもある。第5世代コンピュータは1990年代のコンピュータとして日本が提唱したもので，素子による違いではなく，非ノイマン型の独自に推論，判断が行える人工知能の実現を目指した。→(参)人工知能，非ノイマン型コンピュータ

(2)ソフトウェア面でも，OSやプログラム言語の発展の過程を，世代という表現で区分することがある。→(参)第四世代言語

②プログラムやデータなどのファイルを更新する場合，更新前後のファイルをそれぞれひとつの世代と考え，更新すると世代が代わったという。

③コンピュータシステムをグレードアップなどでモデルチェンジした場合，それぞれひとつの世代ということがある。

世代交替法(generation system)

ファイルを更新するときの世代管理の手法。マスタファイルを更新した後も，前回，前々回のファイルを定めた世代(または定めた期間)にわたって保存すること。誤った処理，ファイルの追加や脱落などの事故にそなえる措置である。

節(node)

木構造で，枝(branch)の集合点を節(node)という。したがって節は枝の分岐点(節点)になる。グラフ理論では点，通信ネットワークでは，局ということがある。→(同)ノード，(参)木構造

設計情報(information for system design)

プログラムの内部設計の段階で管理する情報の集合。多くの技術者が設計作業に係わるため，また検討の結果，変更されることも多いので，常に最新の仕様が一元的に管理され，技術者の端末から登録，修正，照会できることが望ましい。おもな内容は，(1)データ定義情報，(2)データ項目，(3)プログラム情報，(4)ファイルレイアウト情報などである。

セルフアセンブル(self assemble)

実行用の計算機で，ソースプログラムをアセンブルすること。通常のアセンブル法で，自機種用のアセンブラがある機種で可能。ある計算機上で働く機械語を別の計算機上で生成するクロスアセンブルと特に区別するときの用語。→(対)クロスアセンブラ

ゼロ改行(zero inch line feed)

ワープロなどで二重線などによる見え消しや二つの文字を重ね打ちしたい場合に指定する改行指示。上の行にこの指定をすると印刷時改行せず，同じ行に次の行を重ねて出力する。通常のワープロの場合，画面上に重ねて表示できるものは少ない。

ゼロクリア(zero clear)

累算(累加＝足し込み)を行うとき，累算結果を記憶する変数の値を最初に0にしておくこと。プログラムではあらかじめ初期値としてゼロクリアする場合と累算の実行直前に0を代入する場合がある。

ゼロ抑制(zero suppression)

数字を表示するとき，小数点以上(整数も含め)の有効数字(有効けた)より上位にある0をなくすること。ゼロ消却ともいう。例えば0012.3を「12.3」と上位の00を消して表示すること。例えば，COBOLではPICTURE句での数値編集Zがこれにあたる。

線記号(line symbol)

流れ図記号のひとつで，データや制御の流れを表す。実線，破線，通信線などを用いる。必要なら方向を表す矢印を付ける。

全銀協標準通信プロトコル(standard protocol of federation of banker's association Japan)

全国銀行協会連合会が1983年に制定した企業と銀行，銀行間のデータ交換を実現する標準通信プロトコル。データリンクレベルは，BSC手順のコンテンション方式に準拠しているが，上位層に相当するプロトコルも独自に規定している。JCA手順とともに，わが国におけるEDI(電子データ交換)の代表例である。

線形計画法(linear programming：LP)

ある条件のもとで，いくつかの変量を組み合わせて得られる可能解のうちから最適解を求める方法。ORの最も代表的な手法で，需要予測などの本来確定できない値，技術的に正確に観測できない問題，輸送問題・在庫管理などの大規模な問題を，モデルをつくって解く手法。経営工学で広く利用する。代表的な解法には図式解法，シンプレックス法などがある。

セ

線形構造(linear structure)

同一の型のデータが一連に並ぶデータ集合で，その位置を指定して個々のデータを参照できる構造。配列やテーブルがこれに相当する。

線形探索(linear search／sequential search)

配列，またはリストにあるデータを一端から順に探索する技法。配列(リスト)中に，レコード以外のデータ(例えば，空欄やEOF)があるか，レコードの件数は既知か，同じキーのレコードが多重登録されているかなどでアルゴリズムに若干の違いがある。データ量がn倍になれば，探索のためのデータ量もn倍になる。逐次探索，順次探索ともいう。また探索効率の低さを皮肉って野蛮なサーチ(brute force search)ともいう。→(関)番兵

線形リスト(linear list)

データ要素の数が動的に変化する1次元の表で，アクセスがその両端に限られるもの。スタックは線形リストのうち一方の端からのみアクセスできるものである。→リスト，スタック

宣言(declaration／directive typing)

プログラム中で使用する変数や型，関数の記憶クラスをプログラムの初めに定義すること。文には代入文や制御文，入出力文など実際にプログラムを実行する実行文と，実行に先だって変数や関数などの型や大きさを宣言する非実行文がある。言語処理プログラムは宣言を受けて，関係する言語対象物に識別子を与え，属性を割り当て，処理の方式を決定する。例えば，COBOLでは，変数はDATA DIVISIONで定義し，FORTRANでは，実行文に先立ち配列の大きさやデータの型を宣言しなければならない。

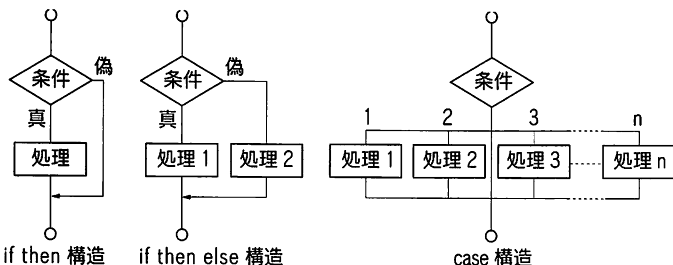
前置演算子(prefix operator)

Cの用語で，オペラントの前に置かれる演算子で，インクリメント(++)デクリメント(--)，アドレス演算子(&/*)，sizeof等がある。算術演算型の前置演算子が評価されたあとでオペランドが評価される。

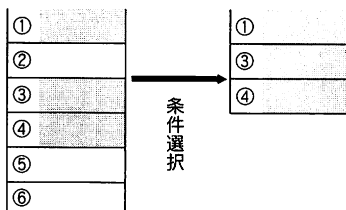
選択(select)

①プログラムの制御構造のひとつ。指定した条件にしたがって，いくつかの文の中から，実行すべき文を選ぶこと。二者択一により処理を実行す

る構造をif構造，三者以上の中からどれかひとつを選択して処理を実行する構造をcase構造という。case構造はif then else構造を多重化したものといえる。



②関係データベースのデータ操作の一機能。元になる表から必要なデータをある条件を設定して取り出す作業。射影は列単位にデータを取り出すが，選択は行単位でデータを取り出す。制約ともいう。



選択ソート(selection sort)

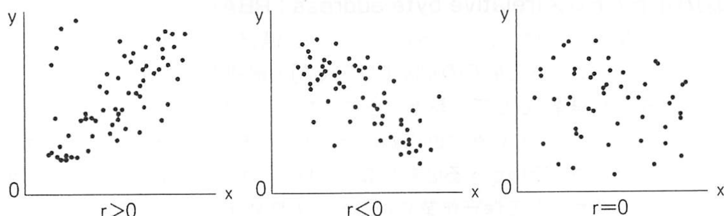
簡単な内部ソートのアルゴリズムのひとつ。配列中のデータの最小値(または最大値)を見つけ配列の一端におき，探索範囲を1件分狭め，同様の操作を繰り返すソート法。バブルソートに比べ交換の回数は減るが，やはりデータ量の2乗に比例する計算量が必要で，すぐれたアルゴリズムとはいえない。最小値法(最大値法)ともいう。

増加テスト(incremental test)

テスト済みのモジュールに未テストのモジュールを結合して，順次に行うテスト。1回のテストで検証されるのは，未テストモジュール内部と結合部分であり，エラーの発生部分の特定も比較的狭い。テストが終了，修正したら，次のモジュールを加えて同様のテストを繰り返す。モジュールの結合の順序によりトップダウン型とボトムアップ型に大別できる。

相関係数(correlation coefficient)

二つの変数x，yの間の関係の度合を表す数値。単相関係数ともいう。相関係数rは常に－1≦r≦1であり，r》0のときは正の相関，r《0のときは負の相関，r＝0のときは無相関という。



総合テスト(integrated test／system test)

開発されたシステムが，総合的に見てシステム設計段階の目的・目標を実現しているか確認するテスト。システムテストともいう。主なテスト項目は，機能テスト・過負荷テスト・安定性テスト・耐障害テスト・性能テストなどであり，運用の現場にシステムが移行された場合を想定して開発者が行うテストである。

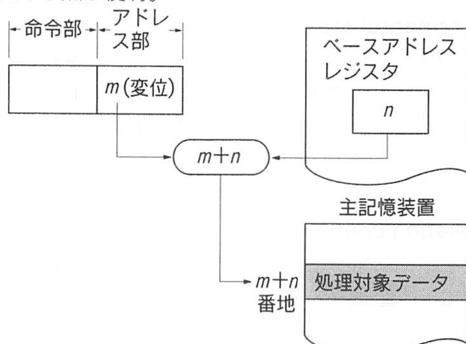
相対アドレス(relative address)

別に指定したアドレスを基準として，相対的に表したアドレス。多重プログラミングで，いくつかのプログラムを同時並行処理する場合，プログラムの記憶領域を絶対アドレスとして固定すると，記憶装置内に空きの領域があってもほかのプログラムを実行できない場合がある。そこで，実記憶装置に格納する段階で相対アドレスから絶対アドレスに修飾(変更)する方法をとることにより，空いている領域を有効に利用し，多重プログラミングを可能とした。→(関)再配置可能プログラム

相対アドレス指定(relative addressing)

命令語のアドレス部に基準値(ベースレジスタ，または命令アドレスレジ

スタが記憶する値)からの変位を指定する方式。プログラムを記憶装置のどこにでも置ける点が便利。



相対バイトアドレス(relative byte address：RBA)

VSAMデータセットで，各レコードの記憶アドレスを表現するアドレス。データセットの最初の制御エリア(CA)の最初の制御インタバル(CI)の先頭番地を基準0として，基準から数えたバイト数でレコードの所在を表す。このようにCI別に相対的に表したアドレスを用いると，シリンダ番号やトラック番号で指定する必要がなくなりハードウェアからの独立性が増し，データセットの保守作業やレコードの移動が容易になる。

相対ファイル(relative file)

相対レコード番号によりファイル内のレコードを識別できるように編成されたファイル。相対レコード番号の昇順に呼び出す順呼出しと，乱呼出しが可能である。なお，一般に他のファイル編成法に比べて大きな記憶容量が必要になる。

相対編成(relative organization)

ファイル編成法のひとつ。ファイル上の最初のブロックを 0番ブロック，次を1番ブロックとし，そのブロック番号を指定してファイルを読み書きできる。例えば，あるファイルの先頭から数えて58番目のブロックを読み出すには，相対ブロック番号を58と指定すると，ソフトウェアがブロックの位置を算出し，該当トラックおよびブロックにヘッドを位置づける。このような乱呼出しができるファイル編成。通常1ブロック1レコード(固定長レコード)であり，この番号を相対レコード番号という。

相対レコードデータセット(relative record data set：RRDS)

相対レコードデータセットは，直接編成ファイルと同様に，レコードのキーをレコードアドレスに変換して，レコードへの直接アクセスが可能な編成

をもつVSAMデータセットである。RRDSでは，データコンポーネント内はスロット(slot)という単位に分解される。スロットは論理レコードを記憶するための器であり，固定長である。スロットには先頭から順に番号が付けられ，この番号でレコードの位置を特定する。この番号を相対レコード番号(relative record number：RRN)という。論理レコードはキー値と対応するRRNのスロットに記録される。このとき，対応するキー値をもつレコードのないスロットは空きの状態になるが，これを空きスロットと呼ぶ。RRDSへのアクセスは，KSDSと同様，順アクセス，直接アクセス，動的アクセスが可能である。また，RRDSに対する処理も，更新・削除・追加が可能である。



相対レコード番号(relative record number：RRN)

仮想記憶アクセス方式(VSAM)で固定長レコードがもつレコード番号。VSAMファイル内の相対編成ファイルに相当するファイルの最初のレコードからの隔たりで，物理的なディスクのアドレスを計算できる。

挿入(insert／insertion)

①データの並び(広く，文字列・配列・レコードの並び，ファイルなどを含む)の中に別のデータの並びを適切な位置にはさみ込むこと，あるいはその機能。追加ともいう。

②すでに入力ずみの文字列の指定位置に別の文字列をはさみ込む機能。または，その入力モード。このモードを特に上書きモードに対比して挿入モードという。

挿入ソート(insert sort)

簡単な内部ソートのアルゴリズムのひとつ。基本的な考えは，配列中のデータを整列ずみの部分と未整列の部分に分割し，未整列のデータを1件ずつ整列ずみの列の中の正しい位置に挿入することの繰返しである。整列ずみの配列については挿入位置の探索に2分探索法が使えるので効率がよい。しかし，挿入のために配列中のデータの番地をずらすことが必要である。データをずらしながら挿入位置を探索するアルゴリズムもある。やはりデータ量の2乗に比例する計算量が必要であるが，データの大半が正順で，追加された一部データだけが乱順である場合などには効率がよい。

挿入法(insertion sort)

ソート済みのテーブルに，新たなレコードをテーブルの並び順にしたがって追加する方法。テーブルの後ろから，ソートキーの比較とレコードの交換を行って，追加レコードの入る位置をさがし挿入する。1件の追加レコードに対して，平均(テーブルの個数／2)回の比較，交換を行うため，処理効率が悪い。

増分(increment／incremental／incrementation parameter)

①一定回数の繰り返しなどを制御する文(FORTRANのDO文，BASICのFOR…NEXT文など)では，繰り返すたびに回数を制御する変数やポインタに加算する一定の値。きざみともいう。増分の値が負の場合，特に減分という。→(対)減分

②レジスタや特定の記憶領域の値に一定の値を加えること，またはその加える値。カウンタやスタックポインタなどに用いる。

ソ

添字(subscript)

配列やテーブル中の特定の要素を識別するために配列名やテーブル名とともに用いる順序数。プログラムの中では，整数または変数の順序数要素をかっこでくくって表す。

ソースプログラム(source program／source code)

プログラマがプログラム言語で表現したプログラム，またはそのテキストファイル。例えば，C，FORTRAN，COBOL，アセンブラなどで記述したプログラム。原始プログラムともいう。コンピュータがこのプログラムを実行するためには，さらに言語翻訳プログラム，連係編集プログラムにより機械語に変換する必要がある。→(関)コンパイル，アセンブル

ソート(sort)

ファイルまたはレコード中の各データ項目を定められた規準(例えばアルファベット順，数の大小，日付順など)に従って順序よく並べ換えること。整列，分類ともいう。順序には，昇順(正順)とその逆の降順(逆順)とがある。並べ換えにより主処理を正しく能率よく行うことができる。そのため前処理としてデータをソートすることが多い。また出力データもソートにより順位がわかる，検索が容易などの理由でソートすることが多い。

ソートの際に大量のレコードを短時間に処理するために，アルゴリズムがくふうされ，各種の手法(例：クイックソート・シェルソート・ヒープソートなど)がある。→(関)外部ソート，内部ソート，昇順，降順

ソートキー(sort key)

各データ項目を順序よく並べ換えるために，指定する項目。通常のレコードは，複数の項目からなり，どの項目で並べ替えるかにより出力が異なる。また複数のソートキーの優先順位を指定できるのが普通である。

社員ファイルで，入社年度をソートキーにすれば社歴の順に，生年月日をソートキーにすれば年齢順に，氏名の読みをソートキーにすれば五十順に並べる。第1キーに入社年度，第2キーを氏名の読みにすれば，全体は社歴の順で，同じ社歴であれば氏名の五十音順に並んだ名簿ができる。

属性(attribute)

①データやファイルに固有の性質，または備わっている性質。例えば，文字種はデータの属性，アクセス権の設定はファイルの属性である。

②データベースの関係モデルを表現した二次元表における列のこと。行のほうは組(タプル)という。

③データモデルのひとつであるE-Rモデルで，実体と結び付く(関連づける)値を定義する性質。例えば，ある人物A(実体)を記述するデータは無数にあるが，その中から(性別，生年月日，身長，体重)という属性に注目すると，人物Aに関連づけた値(実現値)は(男，19741103，171，63)などとなる。→(関)E-Rモデル



ソフトウェア(software：S／W)

不特定多数の人が頻繁に使用するプログラム。一般には大規模で複雑になることが多く，複数人で開発する場合が多い。プログラム・プログラム開発技法・コンピュータの運用法・利用技術やこれらに関する書類(ドキュメント)などを抽象的にとらえたものである。一般には，プログラムやマニュアルそのものをさすことが多い。ソフトウェアはハードウェアと比べ，論理的に複雑な表現ができ，改変しやすく，複製ができるため，データ処理システムの中では工業製品であるハードウェアを多様な目的に利用できるようにする役割をになう。

ソフトウェア危機(software crisis)

ハードウェアの普及とそれにともなうソフトウェア需要の伸びに対して，ソフトウェア生産性の伸びが立ち遅れているギャップから発生するさまざまな問題，またはこれらの問題が引き起こす状況。

(1)大規模化するソフトウェアを体系的に誤りなく開発する手法の模索

(2)ソフトウェア需要に追いつかない技術者不足。システム保守，バックログに追われ新システムの開発に手が回らない開発部門の状況

(3)システムの社会化によりソフトウェア事故の影響の広域化，甚大化

ソフトウェア工学(software engineering)

ソフトウェア開発の過程を対象とし，経済性，信頼性，効率性等の向上を工学的手法により追及する学問。構造化(構造的)プログラミング技法，ソフトウェアライフサイクル，変更可能性(互換性・移植性)，保守性，テスト方法等に多くの貢献をしている。ソフトウェア工学が重視される背景には，基本的にソフトウェア危機がもたらした次の問題があった。

(1)巨大化する規模の問題(大規模ソフトウェアの合理的開発法の要求)

(2)開発量の問題(修得しやすく実践しやすい開発技法の確立)

(3)欠陥の問題(ソフトウェア品質管理とテスト技法)

これらの要求に応えて，1970年代前半に構造化プログラミングの提唱，80年代前半に各種支援ツールの開発，90年代にはプログラムの自動生産技術の実用化と発展してきた。



ソフトウェア使用許諾契約(license agreement／license contract)

ソフトウェア製品を使用するにあたり，メーカーと使用者との間で交わす契約書。一般に，フロッピーやCD-ROMが入った箱や封筒に添付してあり，製品を開封すると契約書に同意したものとみなす。契約書の内容は，使用できる台数や使用条件，複製の可否，第三者への譲渡，保証の範囲，免責，輸出規制，著作権，契約期間など多くの条項を含み，製品の開封前で契約書に同意できない場合は，返品することもできる。

ソフトウェアハウス(software house)

コンピュータ用ソフトウェアの設計開発を専門に行い販売している企業群の通称。コンピュータシステムでソフトウェアコストの占める比率の増加とともに誕生し，発展した。出資先や取引先から，コンピュータメーカ系，銀行系，公共機関系，独立系などに分かれる。

ソフトウェアパッケージ(software package)

注文生産によるソフトではなく，ある機種に合わせて多くのユーザが類似の目的で使えるように考慮して商品化したソフト。大量販売のねらいがあるため廉価で多機能なソフトが多い。なお，ユーザの利用目的に合わせて，パッケージソフトの機能の一部を変更することをカスタマイズという。ソフトウェアパッケージはその内容により，さらに分類することがある。

(1)アプリケーションパッケージ：給与計算，顧客管理，在庫管理などのデータ処理用のプログラム

(2)システムパッケージ：オペレーティングシステム，日本語FEP，表計算ソフトなどの基本ソフトウェア

(3)ツールパッケージ：システム開発支援，システム運用支援，ユーティリティなどのミドルウェア

ソフトウェア品質(software quality)

広義には，ソフトウェア機能(functionality)，ユーザの操作性(usability)，開発者の効率(efficiency)，保守性のよさ(maintainability)，移植性(portability)をさす概念。狭義には，運用時の信頼性(reliability)をさす。ソフトウェア品質の向上のためには，エラーを作らないこと，作ったエラーを発見し除去することが必要で，綿密な設計技法・テスト技法の徹底，品質評価項目と基準の設定が肝要である。

ダイアグラム(diagram)

図表や図式のこと。チャートやグラフを含む。システムの構造やデータの流れや処理の制御構造を表現する(定義する)場合に用いる。最も広範に使われるのはフローチャート(流れ図)で，JISでも規定している。このほか，HIPO，NSチャート，PAD，YAC，ジャクソン図，ワーニエ法など多様な表現があり，それぞれ特色がある。

大域変数(global variable)

プログラムで，各ブロックを超えて，同一名称で使用できる変数のこと。

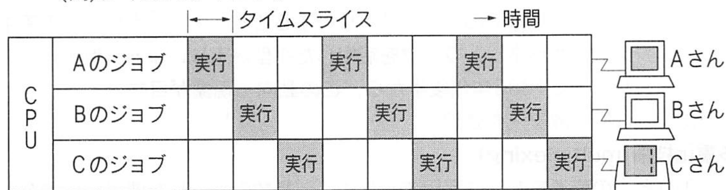
FORTRANでは，COMMON文で定義する。C言語などでは，変数をEXTERNAL宣言して大域変数として扱う。→(対)局所変数

第1正規形(first normal form)

関係データベースでデータを正規化をした最初の表現。すなわち，あるデータベースを構成するレコードが，それ以上分解できないところまで分けられているとき，その表は第1正規形(first normal form)という。ファイルでいえば繰り返しのある項目がなくなりすべて独立した項目の表現になる。→(参)正規形

タイムシェアリングシステム(time sharing system：TSS)

コンピュータの共同利用の一形態であり，マルチプログラミング技法の一形態。ネットワークで接続した複数のユーザ端末が，独自にホストコンピュータと会話する形で処理を行うシステム。時分割システムともいう。ホストコンピュータは，各端末のプログラムに使用時間を小さく区切り(微小時間をタイムスライスという)，順番に割り当てて実行する。コンピュータは多数のプログラムを同時並行的な処理をしているように見える。そのため，ホストコンピュータとセションを確立した使用者は，あたかもコンピュータを独占しているかのような感覚で使用できるという特色がある。→(関)タイムスライシング



タイムスライシング(time slicing)

①広義には，ひとつの装置や回線を複数の処理に利用するために時間を微小時間に分けて配分する方法。データ通信の時分割多重化方式やOSのタスク管理のもとで複数のタスクにCPU時間を配分するときの一方法。

②タスク管理のもとで複数のタスクにCPU時間を配分するときの一方法。CPU時間を各タスクに配分(ディスパッチ)する方式で，タスク優先順位による方法(プライオリティ方式，ダイナミック方式)と均等にCPU時間を配分する方法(タイムスライス方式，ラウンドロビン方式)がある。

タイムスライス方式では，CPU時間を一定の時間に切断(スライス)し各タスクへ配分する。各タスクの応答時間を一定にするようなコンピュータシステムの運用形態に向いている。一定時間が経過すると，実行が終了してなくてもCPU時間を放棄させる。

対話型処理(conversational processing)

広義には，コンピュータシステムとデータや制御情報を交換しながら進める処理。狭義には，人間がコンピュータシステムに次々と指示を与えながら進めるデータ処理。会話型処理ともいう。対話型処理は，大きく次のように分類できる。

(1)システムを利用する人間とコンピュータシステムとの対話
(2)コンピュータネットワークで結ばれたコンピュータ間の対話
(3)コンピュータ制御下にある機械装置とコンピュータとの対話

通常は(1)の人間とコンピュータシステム間の対話をさす。すなわち，コンピュータシステムの操作の一形態で，利用者とシステムとの間で交互に取り交わす投入(入力)および応答(出力)に基づいてデータ処理を行う方式。典型的な例としては，パソコンのワープロ操作，表計算ソフトの操作などがあげられる。機能の選択，データの入力，結果の出力などは，人間の判断と操作を加えながら進展する。

ダウンサイジング(down sizing)

小型化。情報処理の分野では，コンピュータシステムが従来よりも小型化する傾向のこと。あるいは，従来は大型汎用機で行っていたサービスを，より小型な機種でのサービスへ移行すること。大型汎用機依存の集中処理システムからネットワークを活用した小型汎用機による分散処理システムに移行する1980年代後半から，この動向・現象が目立つようになった。→(参)分散データ処理

多重化技術(multiplexing)

①ひとつの資源を有効に活用するために，複数の独立した要求に資源を分

割して利用させる技術。通常，多重化により効率化して使用している資源にはCPU，通信回線，交換機などがある。例えば，CPU時間を時分割して複数のタスクを処理するタイムシェアリングや多重プログラミング，同一の伝送路上に複数の信号をまとめる回線多重化がある。

②ある機能の信頼性を高めるために，複数の独立した資源を用意して障害に備える場合の制御技術。例えば，デュアルシステムやデュープレックスシステム，バックアップ回線の切り換え，バックアップファイルの作り方などに関する技術。



多重化装置(multiplex equipment／multiplexer：MPX)

一本の伝送路を複数の端末装置(DTE)で共有するための装置。マルチプレクサともいう。複数の低速回線のデータを受けて一本の高速回線に送り出したり，逆に高速回線のデータを受けて低速回線に分割する。これにより同一の伝送路に複数の独立した論理的な伝送路ができる。一般にたくさんの低速回線を使用するより高速通信回線の経費が安い場合にに使用する。多重化方式により周波数分割多重化装置(FDM)，時分割多重化装置(TDM)，統計的時分割多重化装置(STDM)がある。

→(関)パケット多重化装置，回線多重化装置，時分割多重化装置

多次元配列(multidimensional array)

配列のうち，二次元以上の配列のこと。

妥当性検査(validity check)

データや命令が論理的に妥当であるか，処理に適した形式であるかをチェックすること。妥当性検査には書式検査，件数検査，限度検査，冗長検査を含む。

ダミーレコード(dummy record)

記憶領域を占有するための実際のデータではない便宜的な空のレコード。直接編成ファイルのレコードはキー順ではなく磁気ディスクのアドレスに不連続に記録されるが，レコードの欠番部分を埋めるために使用する。

(1)直接アドレス方式では，アドレスごとに順次，実在レコードかダミーレコードを書き込んで行く。レコードがなくなれば残りアドレスにダミーレコードを書く。

(2)間接アドレス方式では，まず全領域にダミーレコードを書き込んで行く。その後実在レコードにより算出したアドレスに実際のレコードを書く。

単一ボリュームファイル(single volume file)

ひとつのボリュームにひとつのファイルを格納したもの。単一(シング

ル)ファイルボリュームともいう。

段階的詳細化(stepwise refinement)

構造化(構造的)プログラミング技法のひとつ。プログラム作成の課題をまず抽象的にとらえデータと処理の内容を記述し，次の段階ではその記述をより具体化したデータと処理の記述に分解(break down)する。さらに次の段階でプログラム言語に近い形式でデータと処理の記述を行う。このような具体化と詳細化を繰り返し，最終的にプログラム言語による記述を完成する。この技法によれば，処理の流れを人間的な思考で定義し，次第にコンピュータ処理に準じた形式に変換していくことになり，誤りが少なく，また読みやすいプログラムが完成する。ウォータフォールモデルは，基本的にこの手法に準じたものである。



→(類)トップダウンプログラミング，(参)構造化プログラミング

段階的統合化(stepwise integration)

段階的詳細化の技法で作られた成果物は，大きなシステムの部品に相当するモジュールであるが，これらを結合し，順次大きなシステムにしていく工程をいう。システムを構築するには，単体のモジュールを結合して次第に大きなモジュールにしながらテストとデバッグを繰り返すボトムアップテストと，段階的詳細化と同じ方向で上位のモジュールから順次結合しながらテストを繰り返すトップダウンテストがある。

探索(search)

主記憶，媒体に記憶してあるデータ(レコード)を，特定の項目(コードや名前)をキーとして探すこと。アプリケーションソフトの「検索」にほぼ同義。一般的な探索の技法には，探索範囲をしらみつぶしに探す技法，探索範囲を次第に狭めていき目的のデータに達する技法，データの所在を演算，またはテーブルを参照して求める技法などがある。配列やテーブルなどのデータ探索の代表的な手法には，探索する集合を走査しながら検査する線形探索法(逐次探索法)，探索する集合を二つに分割しながら検査する二分探索法がある。このほか，ある種の演算によって探索するハッシュ法，テーブルを参照するテーブルルックアップの手法などもある。

端子記号(terminate symbol)

基本流れ図記号の一つで，プログラム単位と外部環境との出口，入口を表す記号。プログラムの入口点，出口点を意味する。

タンデム結合システム(tandem system)

CPUを直列につなぎ，それぞれを処理状態に応じ使い分けたシステム。タンデムは3頭だて馬車の意味。主処理の前に前処理を行う装置をフロン

トエンドプロセッサ(FEP)，主処理後の処理を行う装置をバックエンドプロセッサ(BEP)という。



単票用紙(cut sheet)

A4，B4，B5などの一定サイズの印刷用紙。連続用紙に対比する語。

ダンプ(dump)

ある時点でコンピュータ内部または記憶装置の内容を原形に近い形で出力すること。プログラムやデータのチェックに，次の手法をよく用いる。

(1)デバッギングの際に，必要により，主記憶メモリの内容を出力することをメモリダンプ(memory dump)，ファイルの内容を出力することをファイルダンプ(file dump)という。

(2)プログラムのテストランの最中に，指定した時点で出力することをスナップショットダンプ(snapshot dump)といい，プログラムが異常終了した時点で出力することをポストモーテムダンプ(postmortem dump)という。

(3)チェックポイントダンプ(check point dump)は，プログラムの実行過程のところどころで記憶されている処理結果を取り出すダンプで，障害回復のときに利用する。

(4)静的ダンプ(static dump)はプログラムの実行中の特定時点や終了時点でダンプすることで，通常オペレータや監視プログラムの制御下で行う。動的ダンプ(dynamic dump)はプログラムの実行中にダンプすることで，通常プログラムの制御下で行う。

チ

チェーン(chain)

多数の計算機命令あるいは記憶域をつなぎ合わせ，あたかも一つの命令または記憶域として取り扱うことができるもの。連鎖，リンク(link)ともいう。記憶域のチェーンはプログラム作成上のテクニックとして用いる。普通は，記憶域の一部にチェーンをたどるために次の記憶域のアドレス(ポイ

ンタ)を入れる方法を用いる。特に，磁気ディスク記憶装置でシノニムの取扱いによく利用する。

チェックコード(check code)

転送，または伝送したデータの誤り検査のために送信側が付加するコード。冗長検査符号ともいう。データの入力があった場合，受けた側がデータに同様の処理をして，チェックコードと適合するか確かめる。符号の種類や長さや目的によりさまざまに称される。

(1)データが文字で，冗長符号1ビットの場合

パリティチェックが一般的。付加するコードをチェックビットという。

(2)データが数値で基数の1桁の場合

各桁に重みをつけて加算して，末尾の1桁，または基数で割った剰余，または下位何桁かをコードとする。付加するコードをチェックディジットという。各桁の重みが1(各桁の合計)の場合，特にチェックサムという。

(3)データがブロック，またはフレームである場合

一定の演算で算出したコードを用いる。1ビットなら水平パリティチェックが一般的。2ビットなら群計数チェック，1語ならフレーム検査シーケンスという。

また，付加する冗長符号の長さにより，次のように呼ぶこともある。

(1)1ビットの場合：チェックビット(check bit)

(2)数字1桁の場合：チェックディジット(check digit)

(3)文字1字の場合：チェックキャラクタ，検査文字(checkcharacter)

チェックシステム(check system)

データを処理するとき，出力した処理結果が，正確で信頼のおける内容であるかをチェックするシステム。早期に，厳密に，効率的にチェックし，正しいデータに修正する方策を考慮する。

チェックポイント(check point)

トランザクション処理システムにおいて，プログラムの処理中に生じる障害に備え，処理の実行過程を一定の間隔で記憶するように定めたポイント。OSI参照モデルではこれを一般化して同期点と呼ぶ。障害が発生してデータの内容を破壊した場合でも，直前のチェックポイントまでの保存データとチェックポイント以後のジャーナルファイルを用いて，必要なファイルを再現し実行を再開できる。この再実行をチェックポイントリスタートと呼ぶ。→(参)ロールバック

逐次探索(sequential search)

表の形式に並べられたデータの中から，指定した条件を満たす要素を表の左から順に探していく手法。COBOLには，そのためにSEARCH文が用意されている。

中間言語(intermediate language／transaction code)

高級言語と機械語との間に設けた仮想のコンピュータのための機械語。中間コードともいう。高級言語からの翻訳の効率とハードウェアに対する効率の両方を考慮した結果生まれた。高級言語のプログラムはコンパイルの際に，いったん機種に依存しない中間コードに置き換えたあと，機械語(機種に依存)に翻訳する。また，各機種用に中間コードを解釈・実行するインタプリタを用意しておけば，同じ中間コードを異なる機種で実行可能である。Javaはこの方式を採用している。



注釈(comment／annotation／remark／note)

ソースプログラム中にプログラマが覚え書きとして記述する文や行。この記述はコンパイルの際には無視し，プログラム実行時には何の影響も与えない。最近のワープロソフトには印字しない注釈行を書く機能もある。コメントともいう。読みやすいソースプログラムを作るために，モジュールのはじめに標題等をつけるのに使う。

抽出(extraction)

検索の一機能で，指定した条件を満たすデータを検出すること。検索範囲，検索条件を指定し，新たな表を作ることができる。

帳票設計(document design)

入力設計の基本となる設計で，入力原票(原始帳票)の設計をいう。データをコンピュータに入力するためには，アプリケーションプログラムに直接入力するか，フロッピーディスクなどの記憶媒体に変換してから読み込ませなければならない。入力時の基本となる原始帳票(伝票)は次の点に注意して設計する。

(1)わかりやすく必要項目を明記し，重複記入，誤記入をさける

(2)記入順序は左から右へ，上から下へと書きやすくする

(3)関連帳票との記入位置，記入項目を統一し，チェックしやすくする

(4)扱いや保管が容易なファイリングしやすい形にする

(5)キー操作の回数を減らすなど，入力変換しやすい並びにする

→(参)出力設計，報告書設計，(関)入力設計，入力原票

直接アクセス(direct access)

ファイル中の特定のレコードに対し，その記録順序とは無関係に，また

前回アクセスしたデータとは無関係に指定レコードだけをアクセス，処理する方法。順次アクセスに対比される。オンライン処理でのデータ照会業務などを行うときには必須のアクセス法である。データの特定は，レコードのキーの値をもとに行う方法とデータの記憶アドレスを指定する方法がある。→(同)乱呼出し，(参)ランダムプロセッシング

直接アクセス記憶装置(direct access storage device：DASD)

データのアクセスタイムが，そのデータの記憶場所だけに依存し，直前にアクセスしたデータの記憶場所によらない記憶装置。磁気ディスク記憶装置など，ランダムプロセッシング(直接アクセス)が可能な記憶装置の総称。ダスドともいう。

直接アクセスファイル(direct access file)

直接編成，索引順編成のファイルで，直接アクセスが可能なファイルである。直接アクセスとは，必要なデータを記録順とは無関係にランダムに取り出して処理することで，媒体に記録してある順番にデータを処理する順次アクセスに対比できる。→(対)順編成ファイル

直接アドレス方式(direct addressing)

直接編成ファイル作成方式のひとつ。キー項目の昇順に並べた入力レコードを，キー値と等しい番地(磁気ディスク装置内の格納アドレス)に，1件ずつ格納する記録方式。すなわちレコードのキー値をそのまま，シリンダ番号(CC)，ヘッド番号(HH)，レコード番号(R)に当てて直接的にアドレスを指定する。実際にはレコードのキー値をそのまま使用すると無駄な格納場所がたくさんできてしまうので，キー値に簡単な計算をほどこして求めたアドレスや，キー値と格納番地の対比が記録してある参照テーブルから求めたアドレスに格納する間接アドレス方式をとる。

直接編成ファイル(direct organization file)

レコード内のキーをもとに算出したDASD上の番地をアクセスするファイル編成によるファイル。キー項目から計算で記憶場所を算出し，直接アクセスできるため，乱処理に適している。このため記憶はランダムになり，順処理は可能だが能率は低下する。ランダムファイルともいう。

ツ

追加(insert／supplement／addition／append)

データの並び(文字列・配列・レコードの並び，ファイルを含む)の中に

別のデータの並びを付け加えること，あるいはその機能。データを適切な位置にはさみ込むことは特に挿入という。

通信サーバ(communication server)

クライアントサーバシステムの中で，通信方式の変換機能を提供するサーバ。通信方式は通信サービスの種類やネットワークのプロトコルにより多様なので，通信方式の異なるネットワークに接続しようとするときは，変換を要する。

通信処理システム(communication processing system)

情報処理システムと電気通信システムの境界に位置し，VAN業者によるさまざまな通信処理サービスを提供するシステム。

通信ネットワーク(communication network)

端末間を結ぶ機器(交換機や中継機など)と伝送路で構成するデータ伝送のための通信媒体，通信規約(プロトコル)の集合。通信網ともいう。

突合せ(match／matching)

複数のファイルやレコードについて，データや特定のキーを比較して，同じであるかどうかを照合すること。一般に突合せにより，必要データの抽出，編集，マスタファイルの更新などを行う。→(参)照合

ツリー構造(tree structure)

→(同)木構造

テ

定型業務(programmed work／routine work)

手順の定まった日常業務。オフィスでは，定期的に実施する，または処理の方式や手続きが設定されている業務。明快な構造，条件があり，コンピュータシステムに比較的代行させやすい。→(対)非定型業務

ディジタル署名(digital signature)

情報の送信者が本人のみが知る(秘密の)かぎにより情報を暗号化して送信すること。電子メールなどを使用したペーパレスシステムにおける承認印の代わりに用いる。

低水準言語(low level language)

汎用プログラム言語を大きく「機械向き言語」と「問題向き言語」に分けたときの機械向き言語。具体的には，機械語やアセンブラ言語のことをさ

し，コンパイラ言語に対比して低水準言語という。→(対)高水準言語

定数(constant／literal)

プログラム中で用いるデータの値と特性を明示する文字列(綴り)で，固定した値，あるいは変更しない値，データ項目。

ディスクイメージ(disk image)

フロッピーディスク，MOディスクなどのリムーバブルメディアが記憶するデータ全体のこと。個々のファイルにこだわらずボリューム全部を一括してコピー，バックアップするとき，ディスクイメージで扱うという。



ディレクトリ(directory)

①磁気ディスクなどの記憶媒体に記憶したファイルの保存・管理のための記憶部分。ファイル名や大きさ，作成日，記憶位置，参照モードなどの情報を記憶した登録簿で，一般にはカタログともいう。→(同)カタログ

②区分編成ファイルでは，順編成ファイルをメンバというサブファイルに分割し，各メンバを管理している記録をディレクトリ(登録簿)という。

③UNIX系OSやMS-DOSの体系的なファイル管理機能で，多数のファイルを木構造をもったグループに属するものとして扱うときのグループ。グループには任意の名前を付けることができ，これをディレクトリ名という。

ディレクトリ管理(directory management)

ファイルの数が多い場合，1媒体にすべてのファイルを並列に保存すると煩雑になるため，その内容によりグループ化し体系的に管理すること。グループ化の体系(ディレクトリ)を管理すること。UNIX系OSやMS-DOSでは階層的なファイルシステムを利用して，ひとつの媒体のルートディレクトリの下に分岐ディレクトリ(サブディレクトリ)を作り，その中にファイルを整理して保存する。さらに，そのディレクトリの下に分岐ディレクトリを作ることもできる。これらのディレクトリをうまく構成すると，ファイルの管理が容易になり，利用の際の混乱や誤りを防ぐこともできる。→(類)フォルダ

データ解凍(data expansion／data decompression)

圧縮して記録したデータを圧縮前のデータ形式に復元すること。復元，展開，伸張ともいう。例えば，空白文字が50個連続するという情報から，実際に50個の空白文字を生成する処理をさす。

データ管理(data management)

オペレーティングシステムの主要な機能のひとつ。処理プログラムが入出力命令を出すと入出力装置にアクセスする。その際に物理的な入出力装

置の特性を意識しなくてもよいように論理的なアクセス手段を提供する。オペレーティングシステムのデータ管理は，ファイルを中心としたコンピュータシステム内のデータを統一的に管理することが目的である。その働きには，ブロック単位やレコード単位でのデータへのアクセス以外にも，データの編成方法とそのアクセス法，カタログ，位置指定，記憶，検索，保守，データの保護などがある。

→(関)オペレーティングシステム，データセット

データ記憶定義言語(data storage definition language : DSDL)

データの記憶方法を定義する言語で，データ構造を意識した論理的な定義と記憶装置を意識した物理的な定義のインタフェース部分。データ記述言語の一部，または拡張と考える立場，独立した言語と解する立場がある。

データ記述言語(data description language : DDL)

データベースの定義に関連する命令群で，DBMSが規定しているスキーマ定義言語。NDLとSQLが規定する用語である。主として以下のような機能をもつか，ユーティリティプログラムでサポートしている。

(1)スキーマの生成機能

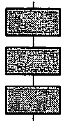
(2)実体ファイル(レコードファイルやインデックスファイル)の作成

(3)実体ファイルのカタログ作成

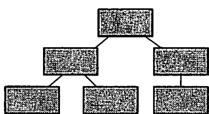
(4)既存データベースの情報提供

データ構造(data structure)

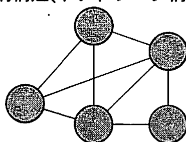
線形構造(配列)



木構造(ツリー構造)



網構造(ネットワーク構造)

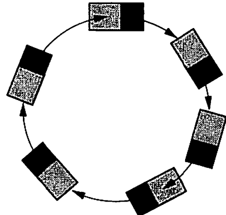


線形リスト構造



データは線形構造と同じ。1列に並んでいるが，入出力はデータの両端のみで行われる各々のデータはポインタで結びつけられている。

環構造(リング構造)



データは線形構造と同じ。最後のデータが先頭のデータとつながっている。

データ要素がどのように関連しているかという論理的構造。データモデルともいう。情報の実体を表現できる構造，アクセス側から制御できる構造，ファイルを構成できる構造などに関連する。線形構造，木構造，網構造，線形リスト構造，環(リング)構造等のモデルがある。プログラム設計に際しては，データの内容，処理方法，記憶方法などを考慮して決める。代表的なデータ構造については，探索，追加，削除などのアルゴリズムの研究が完成している。

テ

データ構造分割法(data structure division)

モジュール分割法の一つで，ファイル構造に対応づけてモジュールに分割する手法。バッチ処理に適している。この分割の図式表現にはジャクソン図が使用されることが多く，ジャクソン法とも呼ばれる。

→(参)モジュール化，ジャクソン法

データ項目(information for data items)

データベースのデータ定義情報の一部で，ファイルのレコード内のデータ項目の名称や属性，長さ，件数，内容説明まで含む。

データコンポーネント(data component)

データコンポーネントにはレコードが入力順に記録されるため，入力レコードはあらかじめ入力順に整列されていなければならない。それぞれのCIには将来の追加や変更などのため空きスペースが作られるが，これは利用者が任意の大きさに設定することができる。

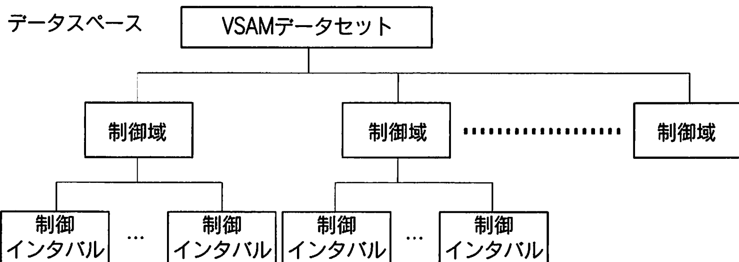
データ処理(data processing：DP)

入力したデータから必要な情報を得るためにデータに施す処理。例えば，計算・ソート・比較・集計・マージ・考察などのこと。処理のための入力，処理結果の出力も含む。情報処理と同義であるが，どちらかといえばデータの自動処理本位の用語である。単に処理といえば，コンピュータ内部のデータ処理であることが多い。

データスペース(data space)

VSAMデータセット用にDASD上に割り当てられた記憶域。VSAMデータセットは必ずデータスペースの中に作成される。また，ひとつのデータスペースには，複数のVSAMデータセットを作成することができる。

データスペースに作られたVSAMデータセットは，いくつかの制御域(control area：CA)に分けて管理される。この制御域は複数の制御インタバル(control interval：CI)によって構成され，この中に複数の論理レコードが記憶される。



データセット(data set)

ある目的のために，関連するデータ項目を一定の並び(入力順またはキー順)で編集，記述したデータの集まり。データを記憶域に格納し，あるいは記憶域から取り出すときに，データ管理が一括して取り扱うデータの単位となる。ファイルと同義で扱われることが多いが，VSAMではデータセットを用いる。→(類)ファイル，(例)キー順データセット，入力順データセット，相対レコードデータセット

データ操作言語(data manipulation language：DML)

DBMSのもとで，データベースのデータを読み書きするための言語。親言語(COBOLやFORTRANなどの汎用プログラム言語)でアプリケーションプログラムを開発する場合(親言語方式)や利用者が直接的にデータを照会する場合(独立言語方式)に使用する。

データチェック(data check)

どんなに優れたプログラムでも，データに誤りがあると正確な処理結果は得られない。そこで，処理に先だち，下記の手法を用いてデータの正確さを確かめることを行う。これをデータチェックという。

(1)字類条件検査

データ項目が英字(ALPHABETIC)か数字(NUMERIC)かを調べる。

(2)論理検査

論理的にあり得ない値が含まれていないかを調べる。例えば，性別コード(男：1，女：2)であれば「1」か「2」以外の数値はエラーとする。

(3)リミットチェック

データ項目の値が一定の範囲に含まれているかを調べる。

(4)フォーマットチェック

データの長さ(桁数)が正しいかを調べる。

(5)トータルチェック

合計したものが手作業で求めた値と等しいかを調べる。

(6)シーケンスチェック

キー項目について，一定の順番に並んでいるかを調べる

(7)バランスチェック

一致するべき項目の値(借方合計と貸方合計等)が等しいかを調べる。

データ抽象(data abstraction)

コンピュータ処理の対象となる実世界のデータを分析し，処理に適したデータ構造を見い出し，グループ化し，データ間の関係を明らかにすること。実世界のデータはさまざまであるが，コンピュータ処理を行うためには，対象となる実体を表現するデータを抽出し，そのデータの属性，構造，性質を考えて処理のアルゴリズムを考えなくてはならない。コンピュータ処理では，基本的なデータ構造についてはデータ処理の方法が確立しているので，処理の対象となるデータ構造と処理のアルゴリズムがわかれば，その部分については基本操作を適用することができる。このような観点からは，プログラミングは，処理対象のデータを，コンピュータシステムで処理可能なデータ型に対応づけることに相当する。

データ定義言語(data definition language：DDL)

データベースの定義に関連する命令群で，DBMSが規定しているスキーマ定義言語。NDLとSQLが規定する呼称で，CODASYLではデータ記述言語(data description language：DDL)と呼ぶ。→(同)データ記述言語

データ定義情報(information for data definition)

プログラムの内部設計に必要な情報の一つで，主としてファイルやデータベースの仕様設計に必要なデータ定義のための情報の集合。

データディレクトリ(data directory)

データベースを定義するスキーマを構成する表，ビュー，権限の定義情報を登録する辞書。カタログ，またはデータディクショナリともいう。データディレクトリへの定義情報の登録はDBMSが自動的に行う。ユーザはその内容を，SELECT文で問い合わせることができる。

データの互換性(compatibility of data)

あるコンピュータのデータが，他のコンピュータ，機器，装置，ソフトウェア等に同一の機能と内容を有すること。データの互換性があればデータ媒体を差し替えるだけで，コード変換をしなくてもそのまま使用できる。

データの互換性が保証できれば，異なる機種で作ったデータを交換したり，まとめたりすることができ，データの流れも連続的になって仕事の能率が上がる。データはそのままでは互換性がなくても，データ変換(code

convert)することで他の機種でも利用できるが，なるべく互換性のある書式で管理することが望ましい。→(参)上位互換性，移植性

データの正規化(normalization of data)

関係データベースで，データの独立性を保つために行うデータのグルーピングのこと。→(同)正規化，正規形

データハイウェイ網(data highway network)

大規模な生産設備(プラント)などの制御を目的に，ミニコン複数台を光ファイバケーブルなど高速通信回線で接続し制御する構内高速通信ネットワークシステム。一台の大型コンピュータで集中管理，集中制御する方式とは対照的に，処理の分散を目的としたシステムである。FA，CIMを構築する場合に必要となる基幹設備。



データバンク(databank：DB)

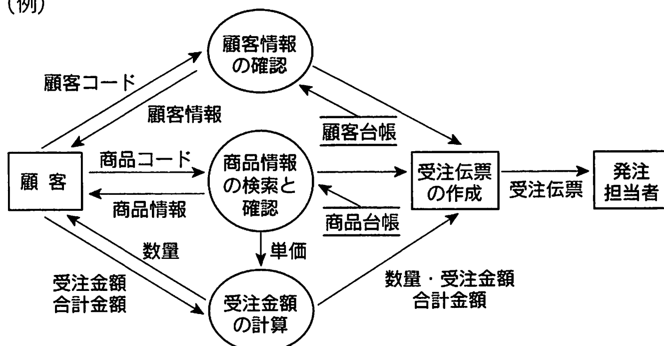
各種データのライブラリをもち，多くの利用者に提供する業態または機関の通称。データ処理の業務で，データについて収集・蓄積・供給・提供・保守の役割を果たすという意味で，データバンクという。また，広い分野にわたって主要な情報が，必要になるたびにすぐ取り出せる形で蓄えている大型のコンピュータシステムをさす場合もある。データバンクの提供例として，マクロ経済データ，株価データ，業界データ，科学技術文献データなどいろいろなデータベースがある。→(類)データベース

データフローダイアグラム(data flow diagram：DFD)

システム要件の定義を行う構造化分析に使用する図表のひとつ。基本的にはシステム間(モジュール間)のデータの流れに注目し，システムとデータの関係を図的にモデル化する技法。この段階では，データの物理構造やハードウェアについては極力触れない。システムの記述を行う場合，次の4つの基本構成要素を使用する。

(1)データの入力元(源泉・発生源・情報源)／出力先(吸収・行き先)：名前を四角形で囲む
(2)処理(プロセス)の内容(処理の機能)：名前をバブル(円形)で囲む
(3)データフロー(データ名とデータの移動)：名前つき矢印で表す
(4)データストア(データ保管物・ファイル)：名前を2本の直線ではさむ

（例）



データベース(database：DB)

相互に関連のあるデータを重複しないように集め，その内容を構造化し，多目的な利用(検索や更新)が効率的にできるようにしたデータの集まり。プログラムにより利用もできるが，プログラムから独立しても利用できる点でファイルとは異なる。昨今は，小型コンピュータの普及，個人ユーザの増加，マルチメディア化の要求の中で，従来のシステムの拡張や簡易化が進み，音声・画像などを含むデータベースシステムも開発され，多様化している。

データベース構築，管理の際には，処理方式や共有するデータの特性とデータ間の関係に注目して構造をもたせるが，この構造，管理の観点から次のように分類できる。

(1)レコード間に階層関係をもたせた階層型データベース

(2)レコード間につながりをもたせたネットワーク型データベース

(3)レコードの集合間に対応関係をもたせた関係データベース

(1)，(2)を構造型データベース，(3)を非構造型データベースという。

データベース管理システム(database management system：DBMS)

データベースとその利用者の間に介在してデータベースの管理を専門に効率よく行うためのソフトウェアシステム。この管理システムにはデータベース記述言語(DDL)やデータベース操作言語(DML)を用意してある。

データベース管理者はこれを用いてスキーマやサブスキーマの定義，データの検索や追加，修正，削除などが可能になる。

・データベース管理システムの役割

(1)利用者が定義するデータの論理的構造について，記憶装置上のデータの物理的構造との対応をとる。

(2)利用者がデータベースからデータを取り出し，操作するプログラムによって，データベースの構造が変更されないようにする。
(3)データを定義するメタデータを，プログラムから独立に定義したデータ辞書(DD／D)で管理する。
(4)データの正常性，整合性が失われないように制御を行う。
(5)好ましくないアクセス，不正なアクセスからデータを保護する。
(6)データベースの利用者にデータベース操作言語を提供する(独立言語方式と親言語方式がある)。
(7)データベース運用のためのユーティリティソフトウェアを提供する

データベース管理者(database administrator)

データベースの設計・運用の知識をもち，企業や機関のデータベースシステムを把握し，管理する専任の技術者。データベースは，データを集中管理し，利用者へのサービス提供を目的とするため利用者とは独立した専門家が必要になる。そのおもな職務は，データベースの設計，編成，保守，運用の監視，障害からの回復などである。

データベース機能(database facility)

表計算ソフトをデータベースソフトとして利用するための機能やコマンド。例えば，データの保守(追加・削除)機能，条件検索，検索したデータを新しい表にする機能，他のソフトへ渡すためのデータ変換機能など。

データベース言語(database language)

データベースシステムを利用するための言語。次の3段階の言語機能をもつ。
(1)データを定義する段階
(2)データを読み書きする段階
(3)得たデータをもとに作表・演算・グラフ表示を行う段階

この全体をひとつの言語処理系でまかなう方式を独立言語方式，(1)，(2)の段階をデータベース言語がになって，(3)の段階を汎用プログラム言語で行う方式を親言語方式という。

データベースサーバ(database server)

クライアントサーバシステムの中で，データベース機能を提供するサーバ。ファイルサーバの一例であるが，特に高速な検索要求に答えるため，キャッシュ機能をもつ。ソフトウェアの例にはアメリカのインフォミックス社のInfomix，オラクル社のORACLEなどがある。



データベース再編成(reorganization of database)

データベースの効率的利用のため，内部スキーマを変更することなく，データの物理的実現方法を変更すること。例えば，アクセス頻度の高いデータを高速アクセス媒体上に移したり，検索経路が短縮するように並べ替えたり，冗長データを削除すること。

データベース操作(database manipulation)

データベースのデータを創成(creation)，再編成(reorganization)，再構成(reconstruction)を行い，読み書き(検索，更新)すること。そのための機能をもつ言語をデータベース操作言語という。

データベースソフト(database software)

アプリケーションパッケージソフトの一種で，簡易なデータベース機能を有するソフトウェア。表計算ソフトや帳票作成ソフト，グラフ出力ソフト，ワープロソフトなどと組み合わせて利用することができる。部門内や個人で利用する小規模なデータベースの構築に適している。

データベース定義(database description)

データベースのスキーマを定義したり，データベースの枠組みを記述したり，変更したりすること。その機能をもつ言語をデータベース定義言語という。DDL，DSDL，DMCLが提供する機能を利用して行う。

データ保護(data integrity)

障害からデータを保護する技術。障害発生の直前までのデータを記録する技術で，定期的にバックアップファイルを作成し，その後の更新記録を残す等でデータを保全する。意識的なバックアップの他，自動的にバックアップファイルを作ったり，ファイルを二重に作ることもある。

データモデル(data model)

データの意味とデータ間の関係を認識・記述するための概念的な構造とそれを操作するための体系。概念的な段階のデータモデルには，(1)階層型モデル，(2)網モデル，(3)関係モデル，(4)オブジェクト指向モデルなどがあり，それぞれのデータ構造に適したデータベースシステムが実現されている。

データレコード部(data record area)

磁気ディスクにレコードを記録する部分。固定長，可変長のいずれでもよい。単一のレコードでもブロックでもよい。末尾に誤り訂正符号を付加する。

テーブル(table)

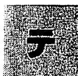
①互いに関係のある項目が，その相対的位置によって識別できるようにデ

ータを配列したもの。その各項目は，一つ以上の引数(添字)またはキーであいまいなく識別できる。表ともいう。

②表計算ソフトや関係データベースソフトで，互いに関係のあるデータを縦横形式の表組になるように配列した表。その各項目は，数式などを用いて一つ以上の引数(添字)であいまいなく識別できる。単に表ともいう。

テーブルサーチ(table search)

特定の項目(キー)を指示することによりテーブルの中から，条件にかなったデータを探すこと。代表的な探索の方法には，(1)線形探索(逐次探索)，(2)2分探索，(3)テーブルルックアップなどがある。



テーブルルックアップ(table look up)

キーとインデックス(指標，例えば記憶番地)となるデータをテーブルに格納しておき，処理時にテーブルのインデックスを参照して目的のレコードの格納場所を探し，処理をすすめること。

テキスト(text)

画像，映像，音声などのデータに対比し，文字によるデータ(表示，印字，ファイルの形態によらず)全般をさす語。情報処理の場では，以下のように使い分けることが多い。

テキストエディタ(text editor)

ディスプレイ画面を利用してプログラムやテキスト形式の文章を編集するプログラム。ワードプロセッサに似た文字列の挿入，複写，移動など機能をもつ。画面全体を使用して編集するスクリーンエディタが主流である。→(類)スクリーンエディタ，(参)ラインエディタ

テキストファイル(text file)

テキスト形式のデータを内容とするファイル。例えば，(1)テキストエディタで編集したソースプログラム，(2)情報交換用のCSV形式のテキストレコード，(3)ワープロ文書の本文データなど。テキストファイルの拡張子には「.txt」，「.doc」などを使用することが多い。

テキストフォーマット(text format)

アプリケーションソフトのデータセーブの一形式。データをテキストファイルの形式で記録すること。テキスト形式のファイルは，他のソフトウェアでも利用しやすく，データの共有，共用に適している。

デシジョンテーブル(decision table)

問題を分析する場合に考慮すべきすべての条件と，条件に対しとるべき処理を列挙した表。決定表ともいう。問題の解を見いだすための説明やプログラム設計の際の補助的な図式として使用する。

条件記入欄

行動表題欄	英数合格	Y	N	Y	Y	N	N	Y	N
	情報処理合格	Y	Y	Y	N	Y	N	N	N
	プログラミング合格	Y	Y	N	Y	N	Y	N	N
条件表題欄	進　級	×							
	進級不可					×	×	×	×
	レポート提出		×	×	×				

行動記入欄

Y=Yes（成立）
N=No（不成立）
X=とるべき行動

デスクトップメタファ(desk top metaphor)

マルチウインドウの環境と機能を利用して，ちょうど机上の作業環境をコンピュータのディスプレイ上に実現したように見せること。メタファは比喩，擬似などの意味。具体的には，ディスプレイに，書類やフォルダ，カレンダーや電卓，ごみ箱などのアイコンを配置し，ウィンドウ画面を文書や書類として何枚も重ねたように表示すること。

デスクリプタ(descriptor)

①レコードのアイテムやプログラムの先頭や末尾に付加する語。そのデータの性質(形式，属性など)を表す。識別子，記述子ともいう。

②データベースの情報検索で，データを検索するためのキーワードとして使用する。情報を分類，索引付けするための名前。

テスト(test)

開発したコンピュータシステムの機能が設計仕様と一致するか確かめること。作成したプログラムの妥当性，システムや装置の正当性の検査。誤りを検出する目的でプログラムを試し実行する。試験，検査ともいう。テストには何段階かあり，システム開発の各段階で計画的に繰返し行う。

(1)設計段階では机上デバッグやウォークスルー

(2)プログラム開発段階では，トップダウンテストやボトムアップテスト

(3)完成段階では総合試験や受け入れ試験

テスト技法は開発技法により変わる。その目的や手法により，名称もさまざまである。モジュール分割したプログラムでは，次のテストを行う。

(1)モジュールテスト(単体テスト)

(2)プログラムテスト(結合テスト)

(3)システムテスト(総合テスト)

またモジュールの結合順序の違いで，次のテストがある。

(1)ボトムアップテスト：下位モジュールから上位モジュールへテストを進める。

(2)トップダウンテスト：上位モジュールから下位モジュールへテストを進める。モジュールテストとプログラムテストの順序が，ボトムアップテストと異なる。

テストのために，トレーサ，デバッガなどの検査プログラムを使う。また，スタブ，ドライバなどのテスト用プログラムを作ることも多い。

テスト計画(test planning)

テストを行うにあたって，立案する計画。テスト計画は，テストの対象となるシステムやテストの段階で異なり，通常，開発部門におけるシステムテストとエンドユーザ部門の運用テストの2段階で行う。



開発部門のシステムテストにおいては，(1)機能テスト，(2)性能テスト，(3)障害回復テスト，(4)負荷テストなどを行う。

エンドユーザ部門では，次の観点から計画しテスト仕様書にする。

(1)関連部門とのシステムの整合性(他システムとのデータ，ファイルの互換性)

(2)オペレータの操作性(伝票・帳票・入力方法など使いやすさ)

(3)並行テストの際は，新旧システムの処理の区分(業務の混乱を避ける)

(4)テスト期間，トラブル時の修正期間，再テスト項目

(5)新システムに切り換え，移行する時点の判断基準

テストケース(test case)

開発したシステムの効率的かつ効果的なテストを行うための必要十分なテスト条件。テスト条件が場合分けされたのが，テストケースであり，テストケースの設計はテスト計画の段階における重要な作業になる。テストケースの設計方法には，次の方法がある。

(1)ブラックボックステスト：プログラムの機能仕様書を吟味して設計する方法

(2)ホワイトボックステスト：プログラムの論理を分析して設計する方法

また，プログラム中のすべてのルーチンをもれなくテストできるようなデータを用意するほか，本来なら実データにないはずの例外的なデータについてもチェックし，システムの信頼性を確保する必要がある。

テスト支援ツール(program testing tool)

プログラムのテスト条件や環境を整備するプログラム群の総称。ドライバやスタブのシミュレータ，テストデータの作成プログラム(test data generator)，広義にはデバッギングエイドがある。

テスト仕様書(test specification)

広義には開発したシステム仕様やプログラム仕様を検査する方法を定義した仕様書。狭義には，特にプログラムの検査法を定義したもの。一般的に次のような内容を含む。

(1)検査する項目，検査法の設計

(2)テストデータの作成法

(3)テストの実行順序

(4)テスト結果の判定法

(5)テストの達成度とシステムの評価

テ

テスト成績書(report of test)

テスト工程とテスト結果を客観的に評価するために，テスト仕様書の定義に従って実施されたテストの記録。実行されたプログラム上の領域，発見されたエラー件数の推移などが記録される。テスト終了の判断を下すときのめやすになる。

テストデータ(test data)

プログラムを検査するために準備するデータ。入力としてありそうなデータからめったにないデータまで組み合わせて，プログラムの全ルーチンを網羅するように適当な数だけ準備する。期待する処理結果も準備しておき，実行させてみた結果と照合する。

テストの目的により，データ量，分布や頻度，境界値や例外値，エラーデータ，ランダム性などを考慮して作成する。

テストプログラム(test program)

コンピュータのハードウェア，ソフトウェアが正常に動作するか否かを検査するためのプログラム。テストのために，特別のルーチンを付加したり，チェックポイントを設けて記憶内容をダンプさせる。

手続き(procedure)

①ある目的を果たすための機械的に実行可能な一連の指令(文)の並び。アルゴリズムにそって記述したプログラムの一つ一つの手順をいう。

②特定の問題を解決する手順(プログラム)に名前をつけて定義し，プログラムのなかで呼び出して実行できるようにしたプログラム単位。手続きが値を返す場合，特に関数という。

手続き型言語(procedure oriented language)

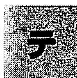
手続きを明確，容易に記述できるような体系をもったプログラム言語。プログラムは処理を指定する文の集合からなり，ハードウェアも原則として記述された文を記述された順序で実行する。命令もデータも同じ記憶装

置内にあり，命令の実行により記憶内容を変更することで処理を段階的(逐次的)にすすめる。FORTRANやCOBOLをはじめ大部分の高水準言語がこれに相当する。

デバッグ(debug／debugging)

プログラム中の誤り(形式エラー，文法エラー，論理エラー，設計エラーなど)を発見し，取り除き，修正する作業。誤りを俗にバグ(虫)という。そこでこの作業を虫取り(デバッギング)ともいう。プログラマがテストの前段階で行う。机上でプログラムを見直して誤りを発見することもデバッグである。文法エラーについては，コンピュータにプログラムを与えて翻訳させ，エラーメッセージを参照して行うのが普通である。論理エラーについては，テストデータを与え，出力をチェックして発見する。

→(参)ウォークスルー，机上デバッグ，トレース



デバッグツール(debug tool／debugging tool)

デバッグを支援するサービスプログラム。デバッガ，デバッギングエイドともいう。トレースプログラム(トレーサ)，ダンププログラム(ダンパ)などを総称する語。広義には，テスト支援ツール(例えば，テストデータジェネレータ)，診断プログラム，シミュレータ(障害シミュレータ，装置シミュレータなどなども含む。バッチ処理で行うものと対話型で行うものとがある。→(関)ダンプ，トレース

デファクトスタンダード(defacto standard)

業界標準。ISO，ITU-T，JISなどの標準化機構が正式に標準として制定または勧告をしていないが，多くの企業がある種の方式に合わせているため，それが事実上の標準となっているものをいう。具体例には，TCP／IPプロトコル，セントロニクスインタフェース，シフトJISコードなどがある。デファクトスタンダードは，後に正式の標準となることもある。例えば，IBMのSDLCがHDLCに，ヒューレットパッカード社のHPIBがGPIB(IEEE-488)に，イーサネットLANがIEEE802.3にと，業界標準が正式の標準となった。

手間(complexity)

アルゴリズムの評価基準の一つで，答えを得るまでの計算に要する実行命令数。同じ答えを得るなら手間の少ないアルゴリズムのほうがよい。計算の複雑さ，計算量ともいう。→(参)時間計算量

デマンド処理(demand processing)

要求時点で即時に行う処理。→(同)リアルタイム処理

デュプリケート(duplicate)

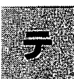
原典から行き先へ，原典と物理的に同じ形でデータを複写すること。事

故や媒体の劣化からデータを守る処置である。例えば，新しい磁気テープに元の磁気テープと同じ形式で複写すること。

電子会議(teleconference system)

複数の会議室を互いにネットワークで接続し，同時に会議を進行できるようにした通信システム。パソコンやワークステーションをホストと接続した電子会議システムが一般的。

電子掲示板システム(bulletin board system：BBS)



通信処理サービスのひとつ。複数の利用者がホストコンピュータのファイル装置内にある掲示板と呼ぶメッセージボックスを介して，各種の情報を提供，交換するサービス。

電子決済(electronic account)

通信ネットワークを介した商取引きで，取引き相手を本人確認し，金銭の決済を行うこと。電子商取引(EC)の基本ではあるが，安全性を考えて慎重な対応が必要との意見もある。利用者は，銀行やクレジットカード会社などの預金から小売店に代金決済する。第三者である認証機関が証明する場合に限り決済が可能になる方式も検討中である。→(関)証明機関

電子商取引(electronic commerce：EC)

通信ネットワークを介した商取引きのこと。電子メール，デジタル署名および電子決済により，ネットワーク上で取引きのすべてを済ませることも可能である。→(類)ディジタル署名，電子決済，(参)電子データ交換

電子データ交換(Electronic Data Interchange：EDI)

取引き関係にある複数の企業間でコンピュータと通信回線を結んで定形業務のデータをやりとりすること。従来のビジネスプロトコル(業界プロトコル)も狭義のEDIといえる。近年ISOによるEDIFACT(Electronic Data Interchange For Administration Commerce and Transport：行政，商業および運輸のための電子データ交換)の標準国内では通産省がCIIシンタックスルールの制定，電信電話技術委員会(TTC)によるISDNとMHSの良さを取り入れたEDI-MHSの標準化など，EDIは業界をこえた標準化という意味でその重要性が認識されるようになってきた。それを受け，1992年 9月に国内の39の業界団体が集合して「EDI推進協議会」を設立した。

EDIの規定は，メッセージのフォーマットをシンタックスルール，標準メッセージ，データコードの 3種類に分けて規定している。各国のEDI標準化は，米国がANSI X.12，欧州がTDIなど，精力的に進めている。EDIで定義するメッセージは複数のデータセグメントで構成し，さらにデータセグメントは複数のデータエレメントから構成する。最近では，企業と消費者との取引(オンラ

インショッピング)を含め，電子商取引，またはエレクトロニックコマース(electronic commerce：EC)と呼ぶようになっている。→(類)電子商取引

電子ファイル(electrically file)

文書や画像などをそのままイメージ情報として電子的に記録し，操作する手段。媒体としては主に，大容量記憶媒体の光ディスクなどを利用する。蓄積されたデータの各種属性を利用して，高速に検索，出力する電子ファイリングシステムがある。

電子マネー(electronic money)

電子技術により金銭の取引きを実現するシステム。デジタルキャッシュともいう。英モンデックスUK社のモンデックスがある。これはクレジットカードサイズのICカードを現金の代わりに利用でき，専用端末でICカードに記録されている金銭の出し入れができる。

電子メール(electronic mail：e-mail)

インターネット，パソコン通信，LANなどのユーザ間で交わすメッセージ，またはそれを実行するシステム。即時性があり，ネットワークどうしが結ばれ，世界規模のメール交換も可能になり，ユーザは急増中である。

電子メールサービス(electronic mail service)

VAN，LAN，パソコン通信などの蓄積サービスの一種。システム内に利用者ごとにメールボックス(郵便の私書箱に相当する)を用意し，メールボックスを介して利用者間で情報の交換を行うサービス。

転送時間(transfer time)

データ転送を開始した瞬間から完了する瞬間までの時間間隔。
→(参)アクセス時間

転送命令(transfer instruction)

主として，主記憶装置と補助記憶装置間のデータの読み書きを指示する命令。入出力装置を制御して直接に，あるいはチャネルを介して間接に主記憶装置と入出力装置の間のデータのやりとりを指示するので入出力命令(input／output instruction)ともいう。特に入力命令を読み込み命令，出力命令を書き出し命令ともいう。事務データ処理においては，使用頻度が高く，処理速度に大きな影響を与える。転送命令は，データ(ファイルの内容，メッセージなど)を送る命令なので，周辺装置の機械制御が必要であり，冗長符号による検査や受信データを送信側に戻す返送照合(転送チェック)を行うことがあり，命令の実行に時間を要する。

テンポラリファイル(temporary file)

①ユーザが，プログラムやデータを一時的に保存しておくファイル。

例)トランザクションファイル
②OSが，あるジョブ内で必要に応じて一時的に作るファイル。ジョブの終了時に自動的に消去する。→(参)スクラッチパドファイル

ト

ト

問合せ(inquiry)

端末よりホストコンピュータに，ある種の処理とその結果を出力するよう要求すること。具体的には，列車や航空機の座席予約状況，銀行の預金残高照会，製品や部品の在庫状態などの問合せがある。なお，問合せによりホストコンピュータが出力した情報を問合せに対する応答(response)と呼び，両者を行うオンラインシステムを問合せ応答システムいう。

動画(movie／motion picture／animation)

映画やビデオ，アニメーションのように動く画像。短時間に多くの画面を次々と繰返し表示すると，残像効果により人間の目は動画と認識する。コンピュータで処理する情報としては，多数の画像データと表示の時間を管理するデータが必要である。コンピュータや通信ネットワークの処理能力を考慮して，画像データを圧縮することが必要である。

動画データ(animation data／video data)

映画，ビデオ，アニメーションなどをデジタル化したデータのこと。動画をデジタル化するとデータ量が膨大になる。そのため，処理装置の高速化だけでなく，データ圧縮，密度の高い記憶媒体，高速伝送などの技術が重要となる。圧縮技術ではMPEG，記憶媒体ではデジタルビデオディスク(DVD)，伝送技術ではATMが今後標準となる可能性がある。

統計的推測(statistical estimate)

統計量より，その標本が抜きとられた母集団の母数を推定すること。

動的アクセス(dynamic access)

直接アクセスと順次アクセスを組み合わせてアクセスする方法。直接アクセスで特定のレコードをアクセスし，それ以降のレコードを順次アクセスにより処理すること。

動的計画法(dynamic programming)

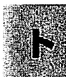
もとの問題を部分問題に分割して解く手法の一つで，小部分の結果を記憶しておき，これを次の段階の問題に適用して新たな結果を求め，次々に

この手法を反復適用し，最終的に全体の解を得ること。この手法が適用できる問題では，状態遷移を考えたときに，逐次決定が確定的である，次段階の問題の最適解が前段階の結果から得られることが必要である。これを最適性の原理(principle of optimality)という。→(参)最適性の原理

動的ダンプ(dynamic dump)

プログラムの実行中にダンプすること。通常プログラムの制御下で行う。→(参)スナップショットダンプ

登録簿(directory)



区分編成ファイルでは，順編成ファイルをメンバというサブファイルに分割し，各メンバを管理している記録を登録簿という。登録簿には各メンバの名称(プログラム名，ファイル名など)や先頭のアドレスなどを書き込んである。似たものにインデックス(索引順次編成ファイルでキーの値で索引が作成する)やカタログ(磁気テープや磁気ディスク上に作るファイルを管理するために作成する)がある。→(類)インデックス，カタログ

ドキュメンテーション(documentation)

文書(ドキュメントやマニュアル)の管理，あるいは与えた主題に関する文書自身の集合。書類または書類中に記録してある情報を作成，収集，編成，保存，引用，配布すること。情報処理の分野では，特にシステム開発に際して，計画書・設計書・仕様書・報告書・運用指示書・保守記録などに関する文書の作成・配布・保存をさす。開発工程のおのおのの段階で必要である。一般に文書化という。

ドキュメント(document)

①広義には，データ媒体に記録した文書およびデータの総称。一般に永続性があり，人間が読み取ることのできるもの。

②システム開発では，開発にともなう文書類・仕様書類を指す。ソースプログラムリストを含む。文書(ドキュメント)を作る作業を文書化(ドキュメンテーション)という。システム運用では，運用にともなう文書類・仕様書類。運用者，利用者向けの操作説明書(特に操作マニュアルともいう)を含む。

独立あふれ域(independent overflow area)

索引順次編成ファイルのあふれ域のうち，各シリンダのシリンダあふれ域からもはみ出したレコードを収容するあふれ域。

独立言語方式(self-contained system)

データベース操作言語(DML)の形態のひとつで，既存のプログラム言語とは独立して，そのDMLだけでデータベースの構築・操作・利用に必要な

処理ができる方式である。利用者は親言語を使用することなく，DBMSに備わっているDMLを用いて対話的にデータを操作できる。この方式は一般の利用者(エンドユーザ)が利用しやすいので，エンドユーザ言語(end user language：EUL)などとも言われる。

閉じたサブルーチン(closed subroutine)

同一処理内容のルーチンをプログラム中に多数回組み込むことはせず，プログラム内に1個だけ挿入しておき，必要な箇所からそのつど呼び出して利用できるサブルーチン。

トップダウンテスト(topdown test)

階層構造化したモジュールの，最上位レベルからテストを始め，順次下位のモジュールを上位モジュールと結合してモジュール間のインタフェースをテストする方法。上位モジュールをテストする場合，その下位レベルのモジュールが未完成の場合，スタブというテスト用プログラムを使用する。トップダウンテストでは，プログラムの中心となる部分(上位モジュール)が下位モジュールとの結合テストにより，繰り返しテストできる利点がある。下位モジュールが未完成のテストの初期段階では，スタブを用いないとプログラム開発と並行してテストをできないという欠点もある。

トップダウンプログラミング(topdown programming)

ソフトウェアの機能をモジュールに分割して，階層構造を明確にしたうえで，上位モジュールから下位モジュールへと段階的に詳細なプログラムを作成する技法。トップダウンアプローチ(topdown approach)ともいう。段階的詳細化技法ということもある。プログラムの全体像を意識したバランスのよいプログラミングが可能になる。プログラムのテストは完成した上位モジュールから行うのが常で，これをトップダウンテストという。
→(類)段階的詳細化，(参)構造化プログラミング

飛越し(jump)

分岐命令によって，プログラムのある部分から他の部分へ制御を移すこと。飛越しを指示する命令を飛越し命令という。→(類)分岐命令

ドライバ(driver)

ボトムアップテストで下位レベルのモジュールまたはモジュールの集まりをテストするために，その上位レベルの未完成モジュールの代替として用いるテスト用のプログラム。テストする下位のモジュールを呼び出して，テストデータを渡す役割をになう。→(対)スタブ

トラック索引(track index)

シリンダ内のレコードを迅速に探索するために，シリンダごとに作られ

る索引。シリンダの各トラックにあるレコードの最大のキーの値とそのトラック番号が記録される。

トラブルシューティング(trouble shooting)

コンピュータシステムに異常が発生したとき，原因をつきとめ，除去，再構築し，システムを正常に戻すこと。またはその作業を支援するソフトウェア。

トランザクション(transaction)

データファイルの内容に，増減，変更を行うすべての取引で発生するデータ。マスタファイルを更新するためのデータ，またはその仕事。オンラインシステムにおけるトランザクションは，通常いくつかの入力メッセージをもち，その応答としての出力メッセージを生み出す。

トランザクションファイル(transaction file)

業務の遂行にともなって新規に発生した一時的なデータを含むファイル。適用業務に応じて，しかるべきマスタファイル(基本ファイル)と突き合わして処理する。マスタファイルに対して入力ファイルとなる。このファイルもまた内容を更新しながら継続して用いる。一時ファイル，発生ファイルともいう。→(対)マスタファイル

トランザクション分割(transaction division)

モジュール分割技法のひとつ。データの流れに着目した代表的な技法。TR(Transaction) 分割ということもある。入力トランザクションの種類(タイプ)ごとに，処理モジュールを作成するもので，特にデータの内容に着目することになる。こうして分割したモジュールは，それに対応する内容のデータのみを処理すればよいことになる。したがって，各モジュールは機能的にまとまったものになり，モジュールの独立性が高くなる。オンラインリアルタイム処理のように，入力トランザクションの種類に応じて処理が異なる場合に有効な分割法である。

トレーサ(tracer)

デバッグやテストの際に用いるプログラムの一種。命令が実行された順番や記憶内容(レジスタや変数の値)および実行結果を出力する。開発中のプログヨムの制御構造や分岐の検査に使用する。追跡プログラムともいう。

トレース(trace)

デバッグの目的で，コンピュータが実行した命令と，その実行によってレジスタの値や主記憶の内容などが変化していく過程を 1ステップ実行するごとに追跡し調べること。

内部スキーマ(internal schema)

3層スキーマのひとつで，記憶装置にデータを格納する場合の記述。この意味で物理スキーマ(physical schema)ともいう。内部スキーマは，蓄積したデータの量や性質，利用目的によって変更されるが，他のスキーマとは独立に定義されていれば利用者には影響がなく，データの独立性が保たれる。

内部設計(detail design)

外部設計(概要設計)をもとに行う，システムの詳細な設計段階。ファイルの仕様，プログラムの構造・機能，入出力帳票や処理手順の具体的で詳細な設計。例えば，入力する原始伝票の設計やプログラム設計書の作成などがある。

内部ソート(internal sort)

ソート処理の過程で，入力ファイルから主記憶装置の容量が許す範囲のデータを読み込んで主記憶装置内部で並べかえを行う作業。ソートの手法にはさまざまなアルゴリズムが考案されている。

→(例)バブルソート，挿入ソート，クイックソート，(対)外部ソート

内部ラベル(internal label)

記録したファイルを識別するため，機械が読み取り可能な形式で媒体内部に書き込むラベル。磁気テープ，磁気ディスク上のファイルの管理や処理を円滑に行うために使用する。→(参)ラベル，(対)外部ラベル

流れ図(flowchart)

問題の定義，分析または解法の図的表現。演算データ，流れ，装置などを表現するために流れ図記号を用いて表した図表。1947年，H.Goldstineとvon Neumannが提唱したといわれている。流れ図を用いることによって作業やプログラムの処理の流れ，使用する資源が正確に把握でき，第三者にも正確に伝えることが可能となる。流れ図は，資源，オペレーション，判断，コネクタなど，一定の形式，順序，要素にしたがって書く。なお，コーディングは，通常，流れ図に基づいて行われる。

代表的な流れ図には(1)プログラム流れ図，(2)システム流れ図，(3)データ流れ図がある。流れ図を作成するときに用いる記号群(flowchart symbol)は，JISが定めている。

流れ図記号(flowchart symbol)

流れ図を作成するときに用いる記号群。

ナップザック問題(knapsack problem)

n 個の物の二つの事柄(例えば，品物の重さと価格，問題を解く時間と得点)が分かっており，いくつかの物を選べるとき，ある制約条件の下で，ある値が最大になる組合せを求める問題。例えば10kg以内の荷が持てるとき，どの品物を選べば価格最大になるか，150分の時間で最大の得点を得るにはどの問題を選ぶかなどの問題のこと。

ナビゲーションサービス(navigation service)

インターネット上の膨大な情報の中から，利用者に必要な情報を絞り込むためのインデックスの提供，誘導サービス，またはその機能をもつソフトウエア。

ナビゲート(navigate)

利用者が見たい情報を検索するとき，案内をするアプリケーションのサービス機能。具体的には，キーワード検索，カテゴリ別に特定のパスにしたがって読み進む検索，最新到着情報検索などの機能がある。

並べ替え(sort／sorting)

→(同)ソート

ニ

日程管理(schedule management)

製品の開発工程と納期の管理。計画には長期的・全体的な大日程計画(master schedule)，中期的・段階的な中日程計画(phase schedule)，短期的・個別的な小日程計画(work schedule)がある。→(参)PERT

日程計画(scheduling)

手順計画や工数計画によって設定された作業時間に基づいて，作業の順序付けあるいは作業開始，終了時刻を工程ごと，作業者ごとに設定することをいう。

入出力設計(input-output design)

外部設計の際に，コンピュータシステムにデータを入力・出力する方法を設計すること。開発部門は，データの用途，目的を十分に考慮し，業務部門の協力を得て設計する。入出力の媒体・タイミング・分量・入手法・配布法などを検討する。特に入出力のヒューマンインタフェースには注意を払い，入力帳票(伝票)と画面レイアウトの設計は見やすさ，操作のしや

すさを考慮して，データ項目を配置し画面定義する。また出力帳票レイアウトの設計では，タイトル・項目の編集・改ページ，集計単位などの処理条件等をよく考えて定義する。→(関)帳票設計，画面設計

入力原票(source document)

コンピュータ処理の対象とする原始データ(source data)を記載した帳票。例えば，取引が発生することで生じたデータは最初は伝票に記録する。この伝票が原始データである。一般に原始データはそのままの形式ではコンピュータ処理に適した形式になっていないので，入力原票(原始帳票＝原票)とディスプレイ画面を見ながらキーボードでディスクなどの入力媒体に変換することが多い。この原始帳票(伝票)は，コンピュータシステムにキー入力する作業のしやすいことが望ましい。入力の基本となる原始帳票のレイアウトは，通常次のような点に注意して設計する。

(1)必要十分なデータを記入でき，重複を避けること
(2)関連帳票との記入位置，記入項目の統一
(3)ファイリングしやすい形にすること
(4)キー入力などの作業がしやすい並びにすること→(関)帳票設計

入力順データセット(entry sequenced data set：ESDS)

順編成ファイルと同様にデータを入力順に記憶する編成をもつデータセット。キー項目に関する順にアクセスする場合は，入力データをあらかじめキー順に整列しておく必要がある。ESDSに対するアクセスは，基本的には順次アクセスであり，先頭レコードからの正順，もしくは最終レコードからの逆順のアクセスとなる。ただし，各レコードの相対バイトアドレス(relative byte address：RBA)を利用者が認識していれば，直接アクセスすることも可能である。ESDSに対する処理を行う場合，更新に関しては更新するレコードが元のレコードと同じレコード長である場合は可能である。レコードの追加は，最終レコードの後ろに追加する場合に限り可能である。削除は，物理的な削除ではなく削除マークを付ける形では可能である。その他の場合は，新しくファイルを作り直す必要がある。

入力設計(input design)

データ入力時の作業に関する仕様を決定すること。入力するデータの項目や量を考えて原始データの帳票，入力装置の画面やキーボードの使い方をくふうし，入力者の使いやすさ，効率性を考えて設計する。入力エラーがあったときの指摘法や再入力のタイミングなども考慮する。
→(参)帳票設計，画面設計

人間工学(human engineering／human factor engineering／ergonomics)

入力装置，表示装置などのヒューマンインタフェースを考える工学。人間が機器を扱うとき，無理なく作業を行うためには，どのように機器を作ればよいかを追求し，操作性や作業環境を考察する学問。エルゴノミクスともいう。→(関)ヒューマンインタフェース

認証(authentication)

データまたは文書が一定の行為，正当な手続きおよび方法，正当な人でなされたことをオンラインデータ上で証明するセキュリティ技術。書類上での認証はサインや印鑑で行うが，オンラインデータはサインや印鑑をそのまま入力することはできない。また，データそのものの改ざんや不正使用も故意か不注意か別にして，特権さえあれば本人でなくても可能である。そのため，オンラインデータの認証を行うソフトウェアを開発している。認証ソフトウェアは，オンラインデータの認証に公開かぎ暗号化技術を応用し，ユーザがそのデータや文書にアクセスできる本人かどうかを証明する。

ネ

ネームサーバ(name server)

ネットワーク内で，ユーザ名やシステム名を保持し，提供するサーバ。クライアントからの問合せにIPアドレスなどを検索して返答するサービスを行う。またはネームサービスを実行するソフトウェア。ネームサーバの具体例として，NIS，DNS，WAIS，WWW，Gopherなどがある。

ネームサービス(name service)

通信処理サービスのひとつ。インターネットのように複雑で構成の変化が激しいネットワーク上のさまざまな構成要素に関する情報を提供するサービス。ここでいう構成要素とは，ノードコンピュータ，ノード上のプロセス，ユーザアカウントなどである。ネームサービスの上位概念で，名前の情報以外のネットワーク構成に関する情報を提供するサービスをディレクトリサービスという。ネットワークプロトコルの下位層では，これら構成要素はネットワークアドレスで識別する。利用者にとっては数字の羅列であるネットワークアドレスで識別するのでは不便なため，ネームサービスが構成要素に対して登録してある名前を利用者の目的に合わせて検索

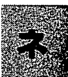
し，その結果を提供する。

ネゴシエーション(negotiation)

折衝の意。通信ネットワークで，通信の前に行う各種の手順。相手の端末の性能，品質，設定状態を見て，通信に不合理なものがあれば相手側にその旨を伝え，お互いに最高なパラメタに設定しあうこと。プレゼンテーション層では，通信相手と折衝(ネゴシエーション)または再折衝し，最適なプレゼンテーションコンテキストを選択する。

ネスト(nest)

関数，反復，選択，複文等の構造で，ブロックの中にブロックを組み込むこと。入れ子ともいう。



ネットスケープ(Netscape communications corporation)

インターネット関連のソフトウェア製品を専門に開発している米国企業。正式にはネットスケープコミュニケーションズ社。代表的なWWWブラウザソフトであるNetscape Navigatorの開発元として有名。1993年マーク・アンドリーセン，シリコングラフィックス社の社長を経験したジム・クラークらがモザイクコミュニケーションズ社を設立。1995年現名称に変更。同社製品で特に有名なのが，NCSA Mosaicを改良，製品化したブラウザソフトNetscape Navigatorである。世界的ベストセラーになり，同社の急成長ぶりが世界中の注目を浴びた。単にネットスケープといえばこのソフトをさすこともある。

ネットワーク(network)

要素の関係を表すデータ構造において，あるデータが複数の上位データ，複数の下位データに関係づけられるネットワーク構造をさす。

ネットワーク型データベース(network structure database)

構造型データベースうち，レコード間の関係が，どのレコードどうしも関連づけることができるネットワーク型(上位と下位の関係がm：n)のデータモデルに基づいたデータベース。CODASYLのDBTGが提案したものが代表的で，CODASYL型データベースともいう。代表的なデータベースには，IDS(米GE社)，AIM／NDB(富士通)，ADBS(日電)などがある。シンコム社のTOTAL，PDM(日立)，DBM(富士通)などもネットワーク型であるが，CODASYL型とは異なるタイプで，部品展開システムの考え方を取り入れており，TOTAL型データベースと呼ばれる。

ネットワーク構造(network structure)

データベース内で，どのレコードどうしも関連づけることができる形のもの。CODASYLのDBTGが提案したものが代表的。上位のレコードに下

位のレコードが複数対応する構造で網の目のようになる。

ノ

ノード(node)

広義には，ネットワークを構成する要素を論理的なモデルとしてとらえたときの呼び方。→(参)節，木構造

①通信システムでは，広く回線や中継線を集中して接続する節点。局ということもある。データの発信点，中継点，受信点などがノードになる。ノード間をつなぐ回線をリンク，またはコネクションという。

→(関)ネットワークトポロジ

②ネットワークを構成するコンピュータ，交換機，中継機，端末，通信制御装置などの物理的な装置。

バージョン(version)

プログラムやマニュアルなどの版番のこと。Version3.0などとバージョンナンバーを付けて，ソフトを更新するごとに版番を上げ，区別と改良と向上の度合を示す。既存ソフトウェアの機能改良や性能向上を，バージョンアップという。通常，最初の公開ソフトをVersion1.0，それ以後ソフトを更新するごとにVersion2.0，Ver2.1などと表示する。版番を示す数字のつけ方には特に規則はないが，部分的な小改訂では0.1を加え，機能を刷新した場合には1.0を加えることが多い。また3桁の英数字で表すこともあり，3桁めの数字はバグ除去程度の修正である。なお，公開に先だって開発メーカ内部のテストに供する版をアルファ版，その後，限られた範囲のユーザの試用に供する版をベータ版という。

バージョンアップ(version up)

ハード，ソフトの性能向上・機能拡張を目的に行う改良。同一のOSが使用できるシリーズマシン，同一のハードで使用できるプログラムなどの場合は，バージョンナンバーが大きくなるほど機能向上や改良の度合が大きいと考えてよい。大規模な改良の場合には，異なるブランド名をつけることもある。

バージョン管理(version management)

プログラムやファイル，マニュアルなどのバージョンを管理し，システム運用中の混乱を招かないようにすること。更新にともなう情報やファイルの管理。運用中のシステムでは，プログラムのバージョンアップやファイルのアップデートを繰り返しながら長期間にわたり利用することが多く，混同は事故や障害を招く場合があり，常に最新のバージョンを利用するべきである。いっぽう万一の事故に備え，更新前のプログラムやファイルを保存しておく必要もあり，多数のファイル情報を正しく管理，利用しなければならない。

バーチャルリアリティ(virtual reality：VR)

コンピュータにより，現実感にあふれる仮想的な空間(バーチャルな空間)を作り出し，あたかも現実であるかのように感じて，観察したり操作したりすることができること。米国航空宇宙局(NASA)が中心に開発した技術で，1989年頃から商品化が始まつた。仮想的な空間に入り込むには，コンピュータが作り出す映像や音が伝わつてくるヘッドマウントディスプレイを頭にかぶり，手の動きをコンピュータに伝える手袋のデータグローブ

を着けるのが代表的な方法。体験ゲーム，スキーやドライブなどのスポーツ，新築予定の家の中の仮想体験などができる。また，飛行機の操縦訓練や原子炉の中や宇宙空間など危険なため人間が入り込めない場所での遠隔作業や訓練などが行える。

パーティション(partition)

ハードディスクの間仕切りをすること。物理的には1基のハードディスクの記憶領域を2つ以上に分割して，論理的に独立した補助記憶装置と見なして使用でき，「パーティションを切る」という。例えば，3つのパーティションに分割したハードディスクドライブには，3つの異なるドライブ名を与えて3基のハードディスクがあるかのように使用できる。

バイナリセーブ(binary code save)

プログラムをコンピュータの内部データ形式で保存(セーブ)すること。一般に，アスキー形式に比べファイルが小さくてすみ，ロードの際にメモリ上に呼び出す時間も短くてすむ。



バイナリファイル(binary file)

①広義には，標準の情報交換用符号によらないファイル。この意味では内容はテキストであっても，文書の暗号ファイルや圧縮ファイル，アプリケーション独自のデータファイルもバイナリファイルである。画像の圧縮ファイルもバイナリファイルであるが，データの互換性を高めるためにJPEGなど標準化も図られており，アプリケーション間での利用が可能になりつつある。

②狭義には，システムの内部表現(機械語や2進コードの形式)のまま記録した形式のプログラムやデータのファイル。入出力時の変換を必要としないため，アクセス時間が最小限ですみ，一般的には記録に要する領域も少ないが，その内容は異機種間での利用は難しく，データであってもコード変換等の作業を必要とする。また，情報交換用符号などのような標準的なコード化をしていないため，内容をそのまま表示しても意味を読み取ることができない。利用する形態のままであり，入出力時の変換を必要としないため，アクセス時間が最小ですみ，一般的には記録に要する領域も少ない。→(対)テキストファイル，(参)バイナリセーブ，ロードモジュール

ハイパーカード(HyperCard)

アメリカのアップル社が1987年に発表したMacintosh用のオーサリングツール。テキスト，グラフィックス，音声などの関連を多次元的に処理できる。ハイパーテキストの考え方の基礎となった。ハイパーカードでは，

情報の基本単位を1枚のページとして1画面のイメージに対応させ，カード間の関係を設定することにより，情報をリンクする。1枚のカードに文字・図形・画像・音声など各種のデータを混在させることが可能である。
→(参)ハイパーテキスト

ハイパーテキスト(hyper text)

情報を項目ごとに分類し，関連する項目を互いに結びつけ，論理的なリンク(関連付け)構造をもつ文書データ。文書データには文字に限らず，図形，画像，音声などの情報を含む。マルチメディア対応のドキュメントシステムである。

ハイビジョン(High Vision／high definition television)

NHKが開発，実用化した高品位テレビ放送の方式。水平走査線数が現行の525の約2倍の1125本，画面の縦横比が9：16と横長である。

パイプライン処理(pipelined architecture)

コンピュータの高速化の一技法。命令処理の過程で，逐次的にではなく複数の命令をオーバラップさせながら処理していく方式。パイプラインのように処理部の入口から次々と命令を入力すると処理部の出口から結果が次々に出てくる。そのときオーバラップしながら処理した分だけ命令処理が速くなる。

ハイブリッド(hybrid)

原義は，混成・混合の意味。技術の分野では，異質な(種類が異なる)複数の仕様に対応していること。例えば，アナログ技術とディジタル技術の両方の長所を取り入れた回路やWindowsとMacOSの両方で実行可能なCD-ROMタイトルをハイブリッドと呼ぶ。

配列(array)

有限個の同じ型の変数の要素を表の形に並べたもの。各要素は，配列名とその順序に付けた添字により参照が可能である。主記憶装置内では，連続した記憶領域内に番地順に記憶される。要素を指定する添字の個数により，1次元配列，2次元配列，3次元配列，……という。1次元の配列は論理的構造と主記憶装置内での物理的並びが一致しており，順次処理で用いることが多い。多次元の配列でも処理系によって，主記憶装置内の1次元の並びに置き換えられる。→(関)行列

バグ(bug)

意図とは異なる処理，誤動作，暴走などの原因となるプログラム中の誤り。バグには形式的な誤りと論理的な誤りがある。虫ともいう。
→(参)デバッグ，デバッグツール

バス(bus)

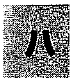
複数の始点の中の任意のものから複数の終点の任意のものへデータや信号を転送する路。CPU，主記憶装置，入出力装置相互間の信号の転送等に使用する共通路。共通路を利用することでシステムの簡素化を図ることができる。複数のプロセッサが複数の主記憶装置を共用する場合もバスで結合する。母線，共通母線ともいう。プロセッサ内部でデータのやりとりを行っているバスを内部バスという。同時に送ることができるビット数により16ビットバス，32ビットバスなどの呼び方がある。16ビットパソコンでは，内部バスは16ビット，外部バスは 8ビット・16ビット。32ビットパソコンでは，内部バスは32ビット，外部バスは16ビットや32ビットなどがある。

バスはその役割により次のように分かれる。

(1)アドレスバスは，プロセッサが要求したデータの番地(アドレス)をメモリへ伝え，また，I／Oへ直接入出力の指示を伝えるもの。

(2)データバスは，両方向でデータのやりとりを行うバスで，プロセッサとメモリでデータのやりとりをし，入力装置(I／O)から入力したデータをメモリやプロセッサへ送ったり，プロセッサ，メモリから出力装置(I／O)へデータを送ったりするもの。

(3)コントロールバスは，プロセッサ内にあるデコーダから出た，書込み・読出し制御信号，タイミング(時間)信号等を各部へ送り，データのやりとりのタイミングコントロールを行うもの。



パス(path)

ファイルをディレクトリという単位で管理する階層型ファイルシステムでは，ファイルはディレクトリの階層の下に存在する。したがって，特定のファイルを見つけるためには，いくつかのディレクトリをたどる必要がある。この検索ルートをパス(経路)という。

ルートディレクトリ(最上位)からのパスを絶対パスといい，カレントディレクトリからのパスを相対パスと呼ぶ。

→(参)ディレクトリ，ファイル管理システム

パスワード(password)

コンピュータ，端末，回線，ファイル等の正当なアクセス権を有する利用者本人かどうかを識別する文字列。合い言葉，暗証ともいう。セキュリティ対策の最も一般的な方式で，システム使用者の認定を行うために，あらかじめシステムに設定した本人しか知らない記号類である。共同利用するコンピュータシステムの使用者は，ログオン時にまず利用者名(ID)を入

れ，それからパスワードを入力する。システムはそれをもとに利用可能者であるか否かを認定する。パスワード自体は，一般にコンピュータの内部またはファイルに暗号化して格納する。銀行や郵便局のキャッシュディスペンサ(CD)で用いる暗証番号もこの一種である。

パスワードは，本人が忘れにくいものにするあまり，利用者名と同じにしたり，生年月日，電話番号といった他人からも容易に判断がつくようなものにしやすい。ネットワークのハッカーが侵入するケースのほとんどは，パスワードに第三者が容易に判断できるものや標準のものを使用していたためである。この点からもパスワードの決め方について十分注意を払う必要があり，パスワードは定期的に変更するほうが安全上望ましい。また，利用者がパスワードを正しく利用するように教育することがシステム管理者の重要な業務のひとつでもある。

パソコンLAN(personal computer local area network)

パソコンどうしで構内ネットワーク(LAN)を構築し，お互いのファイルやプリンタの共有を可能にするシステム。イーサネットと呼ぶネットワークで，クライアントサーバモデルにすることが多い。

パソコン通信(telecomputing／personal computer communication)

家庭や企業内のパソコンで電話回線を通じてホストコンピュータと通信したり，業者から情報サービスを受けたりすること。

電子掲示板(BBS)，メールボックス，電子会議，情報検索，買物の注文，クレジットカードによる決済などのサービスがある。現段階ではプログラムや文字データの通信が主だが，画像・映像の通信も始まりつつある。またひとつのネットワーク内の通信だけでなく，それを入口として他のネットワークにもアクセスできる。パソコン通信サービスの内容はパソコンどうしの通信に加え，電子掲示板，電子メール，データベースアクセス，チケット予約など多様である。利用料金は，接続料金(分単位)とサービスごとに設定した料金とから構成される体系が多い。

また，最近ではパソコン通信ネットワークのゲートウェイ機能の充実により他ネットワークやインターネットとの接続が可能になっている。

パターン認識(pattern recognition)

人間が行う形状・輪郭・構成などの識別を，機械に自動的な手段で行わせること。手書き文字認識，図形認識，音声パターン認識などが実用化している。一般にあらかじめ登録されている多数の標準パターンと入力されたデータのパターンを比較し，最もよく一致するパターンを選出する方式(パターンマッチング方式)が多い。

ハッカー(hacker)

原義(手斧一丁でログハウスを作る人)から転じて，コンピュータ上で人が驚くような大きな仕事ができるプログラマの尊称。さらに原義から遠ざかり，コンピュータにかじりついて離れないマニア的技術者をさす語になった。最近では非合法にコンピュータシステムに侵入し，プログラムやデータの盗用，改ざん，破壊などの悪質な罪を犯す者をさすことがある。しかし，システムに対する犯罪者はクラッカーまたはブレイカーと呼んで区別するべきだという主張もある。

パッキングファクタ(packing factor)

直接編成ファイルでは，媒体に記憶域を確保するとき，シノニムの発生，将来のレコード増加があるものと見て，大きめの記憶域を確保する。実際の記憶域の大きさとファイルに当てた記憶域の大きさの比率をパッキングファクタ，または記録密度(packing density)という。一般に80～85%以上になるとシノニムが急増する傾向がある。

→(関)シノニム，間接アドレス方式，(類)ロードファクタ



バックアップ(back up)

システムに障害が発生した場合に備えて，代行できるシステム資源を用意すること，または実際に代行すること。

①記憶媒体で供給されたパッケージソフトについてのバックアップは，次のようなことがらをさし，単に複製することではない。

(1)コンピュータシステムに障害が発生し，再度インストールしなければならないときに原典，または複製を保管すること。

(2)常時使用するシステムディスクを誤って破壊した場合に備えて，原典を使用せず，複製を使用させるための複製。

(3)ファイルやプログラムの更新に先だって，更新したシステムに完全に移行するまで，現在使用中の資源や環境を保存すること。

②高い信頼性を要求するシステムでは，可能な限りシステムを多重化して障害が発生した場合，短時間に切り換えて代行システムを使用する。

(1)プログラムやデータを予備としてディスク，テープ，フロッピーディスク，ストリーマなどにコピーしておくこと。バックアップファイルという。

(2)ハードウェアの故障に備えるデュプレックスシステムにおける待機系。バックアップコンピュータという。

(3)ネットワークの迂回ルート。バックアップ回線という。→(関)ミラーリング，デュアル構成システム，デュプレックス構成システム，UPS

バックアップ回線(backup circuit)

通信ネットワークで，通常使用する回線に障害が発生した場合に，つなぎかえるために用意する回線。例えば，専用回線に障害が発生して使用不能になったとき，ISDNをバックアップ回線として切り替えるというような方法をとる。障害回復までは，スピードが多少遅くなったり，料金を別にとられることがあるが，まったく使用できなくなるという最悪の事態は避けられる。

バックアップ機能(automatic back up)

システムやアプリケーションに入力がないまま一定時間が経過すると，自動的に作業ファイルやシステムファイルのバックアップファイルを作成し，トラブルに備えるソフトウェアの機能。ユーティリティソフトの機能のひとつで，ユーザーが指定したファイル処理の実行に先立って，もとのファイルを自動的に保存しておき，操作ミスに備える機能をさすこともある。

ハ

バックアップコンピュータ(back up computer)

システム異常時にその機能を代行するコンピュータ。長時間のシステムダウンが許されないような高信頼性システムで使用する。デュプレックスシステムでは，待機系コンピュータともいう。EUCでは，ハードウェアの故障中，修理中に使用できる代替のコンピュータをさすこともある。

バックアップファイル(back up file)

コンピュータシステムでプログラムやデータに異常が発生した場合，修復できるようにデータを保存しておくために用意するファイル。トータルセーブファイルともいう。ファイル障害が発生した場合は，ハードウェア上の障害ならディスクの交換などを行ったあと，最新のバックアップファイルの内容をコピーする。そのあと，バックアップファイルを作成した日時以降，障害が発生するまでのジャーナルログを使用して復旧する。
→(参)復元，チェックポイント，世代交替法，アーカイブファイル

バックエンドプロセッサ(back end processor：BEP)

①複数のプロセッサからなる機能分散型のシステム(タンデムシステムなど)で，通信制御などの前処理を分担するフロントエンドプロセッサ(FEP)に対し，その後の中心的な処理を行うプロセッサをバックエンドプロセッサ(BEP)という。→(参)タンデム結合システム

②メイン処理とは別にデータベース処理専用のプロセッサなどを接続している場合は，それをバックエンドプロセッサという。FEPやBEPは相互に独立して処理を行い，磁気ディスクなどを介して情報交換をする。

バックグラウンド処理(background processing)

タイムシェアリングシステムや多重処理システムの環境のもとで動くコンピュータは，平常時には余剰能力をもつ。この空き時間を利用して，低優先順位でコンピュータを活用する処理。例えば，言語処理，バッチ処理，プログラム開発などに使う。パソコンなどでは，ワープロソフトで文書編集を行うバックグラウンドで，印刷処理を実行する。

→(対)フォアグラウンド処理

バックマン線図(Bachman diagram)

DBMSの最初の商用システムIDS(GE，1963)を開発したバックマン(C.W.Bachman)の提案した，レコード間を関連づける記法。基本的には，実体(entity)と実体間の関係(relation)，これらの集まりである実体集合の概念に基づく。データベースの設計は，実世界のデータの関係を記号で模式化する。長方形はレコード型のデータの集合(実体集合)，矢印はレコード間の関係を示すもので親子集合型(set type)という。これはリンク，またはポインタと考えてよい。レコードの内容自体(入出力の単位となるデータ。データベースでは実現値(instance，またはoccurrence)という)は円で囲む。

バックログ(backlog)

ソフトウェア開発の需要の大きさに比して供給の能力が及ばないとき生じる，未開発のプログラミング業務。例えば，開発ずみのシステムの保守に追われて，要求のある新システム開発に手が回らない場合，新システム開発がバックログになる。

ハッシュ関数(hash function)

ハッシュ法に用いる関数。ハッシュ関数は，キーの分布の偏りを考慮したうえで，ハッシュ表に無駄を作らず，しかもできるだけ均等に配分する関数が望ましい。例えば，キーが整数で均等に分布していれば，mで割った余りは0からm－1までの整数になり，m個のグループに均等に配分できることになる。→(関)ハッシング

ハッシュ合計(hash total)

データチェックの一手法。レコード，またはファイル中の特定の複数の項目の和で，その合計自体は特に意味をもたない。入力前に人手で算出し，データとともに入力し，システム側の算出値と比較することで入力データの誤りの有無を検査する。例えば，同一レコード内の金額とコード番号の和，レコードごとの個人番号の和など。

ハッシュ法(hashing／scatter storage method)

表と関数を用いたレコード探索法の一つ。データを記憶する段階で，キー値から演算により格納場所を算出し，その場所に記憶させる。データを探索する際には，キー値に同じ演算を施せば格納場所が得られるという高速の探索法である。問題点は，演算のルールしだいでデータ量より多くの格納場所をあらかじめ確保しなければならないこと(記憶装置の資源の無駄がある)，異なるキーの演算結果が一致して(衝突またはシノニムの発生という)，記憶場所の競合が起こり得ることである。競合する場合は，同じ場所にあてがわれたレコードをポインタでつないでおく。これをチェーンという。キーから格納場所を算出する関数をハッシュ関数(hash function)といい，データの格納場所をハッシュ表(hash table)という。

ハ

ハッシング(hashing／hash addressing)

直接編成ファイルで，レコードのキーからレコードアドレスを定める方法にハッシュ法を用いたとき，ハッシングという。レコードキーとレコードアドレスは 1対 1で対応すれば理想であるが，一般的には，重複(シノニム)ができたり，空き領域ができたりして記憶装置の使用効率が低下する。そこで多対 1の対応でアドレスを求め，同じアドレスに当たったレコード群をバケット(bucket)に収納する方法をとる。このバケットに割り当てられるレコード数をできるだけ一様にするため，ハッシュ関数を用いる。→(参)シノニム，アドレス生成，直接編成，ハッシュ関数

パッチ(patch)

もとは一時的なつぎ当ての意味。アセンブルやコンパイルの済んだオブジェクトプログラムのバグを手直しする当面の処理。

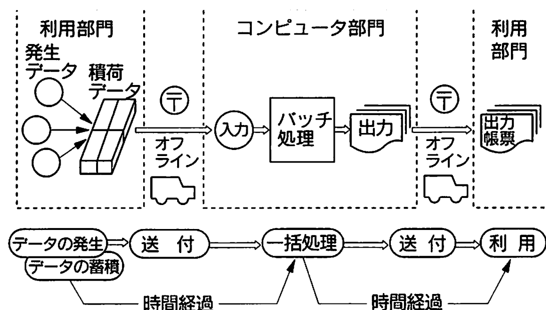
①コンパイル結果やデバッグ結果をとりあえず見たいときなど，再翻訳作業を省いてメモリ上で部分的な修正をし，実行させること。また，ソースプログラムの一部だけを手直しすることも多い。こうしてデバッグした後は，もとのソースプログラムを修正し，流れ図やプログラム仕様書も修正しなくてはならない。これを省くと文書との対応がとれず，後の保守で余計な時間がかかることが多い。

②プログラムの移植に際し，実行不能なプログラムやモジュールを部分的に修正すること。

バッチ処理(batch processing)

データ処理の一形態で，データをある一定の分量，または一定の時点になるまでためておき，一度にまとめて順に処理する方式。一括処理方式ともいう。バッチ処理は処理手順が確定しており，処理の対象となるデータ

もファイルの形式ですべてそろっている場合に適用される。例えば，毎月1回の給与計算，各種料金などの支払いなどの月次処理，企業会計などの年次の処理，また，全ソースデータが揃った時点とか，在庫数が定めた発注ロットに達した時点の処理はバッチ処理方式で処理する。



幅(breadth)

木構造では，一般にルートから隔たるほど頂点の数が増える。ルートから見て同じ深さにある頂点を同一レベルの頂点といい，その頂点の数を幅という。

幅優先探索(breadth first search)

木構造のグラフのデータ探索法の一つで，ルートからスタートし，同一の深さにあるすべての頂点を探索し，なければ次第に深いレベルの頂点を調べる探索法。探索に成功したときは，最短経路が記憶されている。最も深い葉(木の高さのレベル)に至ってもデータがなければ失敗である。

→(対)深さ優先探索

パブリックドメインソフト(public domain software：PDS)

①政府機関，大学や研究所など公的な機関が開発したソフトウェアの一種。ソースコードやソフトを開発した詳細な仕様を公表することが通常で，その著作権や特許権などの使用も許諾することが多く，第三者が自分の目的のため使用することができる。公的資金で基本ソフトウェアを充実し，共用によって社会全体としてのソフト資産や技術力の強化を意図している。

②広義には，ソフトウェアの著作者の意思，意図により，広い範囲に提供したソフトウェア。公開ソフトウェアともいう。日本では著作権法でソフトウェアの開発者に著作権が帰属するため，著作権については著作者の意思表示が必要である。パブリックドメインソフトは，次のような形

態で配布されている。

(1)フリーソフト：無料で使用できるソフトウェア。この意味でフリーソフトと呼ばれる。知的所有権は開発者にあり，改造，再配布などは開発者の指示に従わなくてはならない。

(2)シェアウェア：無料で配布するが，一定期間試用し気に入ったら開発団体へ寄付金，または開発者へ使用料を支払わなくてはならない。

(3)その他：公共機関や教育機関の使用に限り無料，他のユーザは有料とするもの(シェアウェアの例外に相当する)など。

バブルソート(bubble sort)

最も簡単な内部ソートのアルゴリズムの一つ。配列中のデータの隣り合った二つを比較し，逆順であれば入れ換える。これを何回も繰返し，ソートを完了する。簡単ではあるが，データ量が多くなるとその2乗に比例する計算量が必要で，すぐれたアルゴリズムとはいえない。隣接交換法ともいう。

ハミングコード(Hamming code)

誤り訂正符号の一種。この符号方式を1950年に考案したR.W.Hammingの名前をとって名づけた。これにより誤りの検出と訂正を行う方法がハミングコードチェック。基本原理は，送信側が検査ビット(パリティビット)を元のビット列のある部分と組み合わせると一定の規則になるような位置に埋め込む。受信側がその規則と整合性をチェックすると，誤りのあったビットの位置が分かり，そこを反転すれば元のビット列を復元できる。

この符号は，検査(冗長)ビットt個に対して，(2^t-1-t)個の情報ビットの中に発生した1ビットの誤りを訂正できる。また，検査ビットにさらに1ビット付加すれば(拡大ハミング符号)，上と同じ数の情報ビット中に発生した2ビットの誤りを検出できる。

パラメタ(parameter)

①ある用途をもち，一定の値が与えられる変数。例えば，数式 $y=ax+b$ ではa，bがパラメタである。

②プログラム内で，手続きや関数の内容を定義する変数で，プログラム間の連絡をとるために，別のプログラムに引き渡す値。ALGOL，BASIC，Cの用語で引数と同義。

バランスチェック(balance check)

もとは会計処理の貸方合計と借方合計の一致などをチェックする方法。転じて二つのデータが均衡しているか調べて，誤りの有無をチェックすること。平衡検査ともいう。

バンキングシステム(banking system)

銀行業務を自動化するシステム，金融情報システムのこと。主に，銀行内部の事務自動化，銀行間決済，顧客と銀行間の取引きの自動化を行うシステム。オンライン化の流れを時代別に見ると3つに分けることができる。

(1)第 1次オンラインは1965年頃からの，普通，定期，通知，定積などの預金や為替など個別種目のオンライン化

(2)第 2次オンラインは1974年頃からの，当座，貸出，為替を含む銀行業務の即時処理総合オンライン化

(3)第 3次オンラインは対外的ネットワークの拡充が特徴で，金融機関相互，企業と金融機関の間(ファームバンキング)，家庭と金融機関の間(ホームバンキング)で資金決済を可能にする。

→(参)ホームバンキング

ハングアップ(hang up)

障害。コンピュータが何らかの理由で正常な動作をしなくなり，通常なら受け付けるキーボードからの情報を受け付けなくなってしまう状態。もとは，通話がすんで電話の受話器をもとの位置に戻す切断を意味していた。

判断記号(decision symbol)

流れ図記号の個別処理記号のひとつで，ひとつの入口と，複数の(しかし択一的な)出口を表す記号。横長の菱形記号で，記号中に示した条件を評価して，唯一の出口を選ぶ判断機能，スイッチ機能，分岐機能を表す。

判断表(decision table)

→(同)デシジョンテーブル

判断命令(decision instruction)

条件付き飛び越し命令と分岐命令。

番地呼出し(call by address)

手続きや関数を呼び出して使用するとき，パラメタの引渡しを仮引数のアドレスやポインタで与え，実引数との対応をとること。

反転表示(reverse／inverse／negative)

ディスプレイ画面上の文字を目立たせる強調表示法のひとつ。通常の文字と濃淡の逆転した文字を表示すること。例えば，通常の文字が白地に黒で表示するなら，反転表示では，文字のバックに黒帯を作り文字を白い線で表示する。

ハンドル(handle)

グラフィカルインタフェースで，ユーザーが選択したオブジェクトを囲むように表示する黒い小さな正方形の連なり，またはそれらを直線で結んだ囲み。ユーザーはこの正方形をドラッグして位置や大きさを変更できる。

バンドル(bundle)

ハードウェアとソフトウェアの価格を分離しないで販売すること。主にパソコンの分野で，ハードとソフトのセット販売，ユーザによるインストール作業の省略，ソフトの不法複製を防止する販売法として普及した。周辺機器を購入するとドライバやユーティリティが付属してくる場合，主ソフトウェアに補助ソフトウェアやデータを付加して販売する場合でも使うことがある。

販売管理システム(sales management system)

販売管理は，商品の販売に関する情報(商品，市場，仕入れ，在庫，販売，配送，請求，回収，顧客などの情報)の管理全般をさす。狭義には，マーケティング管理をさす。すなわち，市場把握，販売商品の決定，販売組織の確立など販売力の強化をめざす管理である。販売時点情報管理システム(point of sales：POS)は，電子式金銭レジスタ，バーコード読取り装置，クレジットカードの自動判別装置などの機器により収集した商品の販売情報などをコンピュータに送り，情報の有効利用を図るためのシステム。EOS(electronic ordering system)は，受発注業務を自動化したシステム。POSシステムと並び，商品の迅速な仕入れ，販売活動を支援する。顧客のニーズの多様化により，商品の多品種少量生産が進み，受発注業務が煩雑化する傾向にある。これをコンピュータネットワークを利用し，簡便化，高速化を図るねらいがある。

販売時点情報管理システム(point of sales：POS)

→(同)POS

番兵(sentinel)

線形探索で探索の終了時を検出するための手法の一つ。配列(リスト)の末尾にあらかじめ，探索するキーと同じ値(番兵)を入れておき，探索中，キーが一致すれば探索を終了する。探索に失敗した場合でも，最後には探索を終了する。番兵を設けると，データ件数を数えて終了時を判断するルーチンが不要になる。

汎用コンピュータ(general purpose computer)

事務計算，科学計算あるいは経営計算の区別なく，広い範囲の問題処理

や利用が可能なように設計したコンピュータ。ある特定の適用分野向けのために設計した特殊用途コンピュータと対比する語。

汎用プログラム言語(versatile programming language)

プログラム言語のうち，比較的広い分野で多目的に使用できる仕様をもったプログラム言語。アセンブラやBASIC，Cなどの高水準言語をはじめ，多種多様なものがある。比較的，限られた分野で使用される特殊問題向き言語に対比する語。

ヒ

ピアツーピア(peer to peer)

米IBMが最初に提唱した，まったく上下関係のない分散処理システムのモデル。ピアは対等な相手の意。クライアントサーバモデルではクライアントとサーバというように役割が分かれているが，ピアツーピアのモデルではそれすらなくすべてのコンピュータが対等な地位にある。専用のサーバを置かないパソコンLANの形態をとったピアツーピアモデルのネットワークOSもある。

ヒープ(heap)

2分探索木の構造を配列に対応づけるとき，その特性を表す語。配列ａの添字ｉとｊの要素ａ(ｉ)とａ(ｊ)の間に，ｉ＝[ｊ／2](ｉはｊ／2の整数部)であれば，ａ(ｉ)≧ａ(ｊ)が成り立つとき，配列ａは，ヒープであるという。添字とデータの間の大小関係が存在することで，高速なソートのアルゴリズムに応用される。

比較演算子(relational operator)

プログラムやマクロ命令中で，数字や文字の大小比較をする条件などを表すとき，二つのオペランドの比較を関係式に書くための演算子。プログラム言語等によって異なった記号を用いる。関係演算子，比較作用素ともいう。ほとんどのソフトでは次のような演算子で比較ができる。

比較関係	数学記号	演算子の例		
小さい	＜	.LT.	LESS THAN	＜
小さいか等しい	≦	.LE.	NOT GREATER THAN	＜=
等しい	＝	.EQ.	EQUAL TO	＝，＝＝
等しくない	≠	.NE.	NOT EQUAL TO	!=
大きい	＞	.GT.	GREATER THAN	＞
大きいか等しい	≧	.GE.	NOT LESS THAN	＞=

→(関)条件文，条件式，単純条件，複合条件

比較命令(compare instruction)

コンピュータ内部で二つのデータ項目の値の大小(数値の大きさだけでなく，文字列の前後関係などを含む)を比較する命令。また，比較の結果により，コンディションコードを設定する。計算機プログラムの実行中に，複数の命令からひとつを選ぶように指示する分岐命令(branch instruction)の実行に先だって実行される。演算の結果により条件コードの値をセットし，その値により飛越しを発生させる。判断命令，条件つき飛越し命令と同義。条件コードによらない分岐命令は無条件分岐命令(飛越し)命令という。

光の三原色(three primary colors of light)

適当な配分により任意の色を表現できる色光の3要素で，赤(Red)，緑(Green)，青(Blue)の光(RGBと略称)をさす。カラー受像機，カラーディスプレイには，RGBの3色について明度信号を送って色表現を行う。画像作成ソフトなどで混色を行う場合，光の三原色の混色は人間の直観になじまないためカラーパレットなどを用意して，色を選択するようにするケースが多い。

引数(argument)

サブルーチンや関数副プログラムを呼び出すときにプログラム内で引き渡すデータやパラメタ。→(同)パラメタと同義。

サブルーチンを定義するときに使用する引数を仮引数(dummy argument)，サブルーチンを呼び出す側で使用する引数を実引数(actual argument)という。

引渡しテスト(delivery test)

→(類)受入れテスト

非実行文(non-executable statement)

データの型や値を宣言したり，関数を定義したり，記憶領域を確保するなど，処理を行わせる方法を定義する文。例えば，COBOL言語では，見出し部，環境部およびデータ部の記述が，FORTRANでは宣言文，DATA文

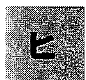
などが非実行文である。→(対)実行文

ビジネスプロトコル(business protocol)

業界内情報ネットワークにおいてデータ通信を円滑に行うためにその業界で定めたプロトコルの総称。全国銀行協会が定めた全銀協標準通信プロトコル(全銀手順)や日本チェーンストア協会が定めた取引先データ交換標準通信制御手順(JCA手順)などがある。なお狭義のプロトコルとしてEDIのメッセージフォーマットのみのことをさす場合がある。

ビジュアルプログラミング(visual programming)

GUI環境下で使用する視覚的な画面と処理の遷移を容易に設計，指定できる機能がある言語プロセッサ(言語処理系)でプログラミングすること。ビジュアルプログラミングができる言語プロセッサでは，通常のプログラム記述が可能なほか，エディタ画面上でのアイコンの表示やマウスの操作の指示を画面上のツールを用いて指定すると，インタプリタとして動作したり，コード化したりする機能がある。パソコンでは，Visual Basic，Visual C++，Delphi等がよく使用されている。



非修理系(unrepaired system)

一度故障すると修理ができず寿命が尽きるシステム。例えば通信衛星のように，故障しても簡単に修理に行くことができないシステムをさす。このようなシステムの信頼性を上げるには，装置の多重化が避けられない。なお，非修理系では，故障の発生確率がランダムの場合，システムを二重化しても平均寿命は1.5倍にしかならず，システムの平均寿命を 2倍にするには 4重化が必要となる。

ビッグバンテスト(big bang test)

すべてのモジュールが完成し，モジュールテスト(単体テスト)が終了した後，全モジュールを結合したうえで行うテスト。未完成モジュールがあればドライバやスタブを用意する。単体テストは終了しており，テストの対象はモジュール間インタフェースに限定できるが，モジュールの数が多いと原因追及の負担が大きい。大規模システムでは，この前段階で結合テストを行う。ビッグバンテストは比較的小規模なシステムのテストに適している。ビッグバンは宇宙創成の最初にあったという爆発のこと。これに例えてシステムの創成時に行うテストをこのように呼ぶ。非増加テストともいう。→(対)増加テスト，(参)結合テスト

非定型業務(non-programmed work)

手順の定まっている定型業務に対し，新たな意思決定や問題解決が必要な業務をいう。業務の革新，開発などに伴い発生する。コンピュータ処理

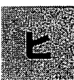
に関しては，定型業務は全社的で自動化，システム化しやすいのに対し，非定型業務は個別的で，部門ごとの処理になりがちでデータの流れや処理に人手を要することが多い。→(対)定型業務

ビデオオンデマンド(video on demand)

映画などのビデオデータベースに端末からアクセスして鑑賞するシステム。長時間の画像・音声データを一度に送っては再生時間同様の長時間を要するため，圧縮データをリアルタイムで送り，十分な品質で再現する技術が必要である。→(関)オンデマンドサービス

非ノイマン型コンピュータ(non-von Neumann type computer)

ノイマン型コンピュータの欠点を克服するアーキテクチャをもつコンピュータの総称。並列処理や分散処理によって処理を高速化・分散化を図る。現在の主流であるノイマン型コンピュータは，プログラム内蔵方式・逐次制御方式を採用しており，一時には主記憶装置にある一命令を逐次に処理・実行する原理のためハードウェア的な処理速度の向上には限界がある。非ノイマン型コンピュータはノイマン型コンピュータの欠点を克服しようとするコンピュータで，多数のCPUを同時に用いる並列処理コンピュータ，連想記憶や論理記憶を使ったコンピュータなどがある。



ビュー(view)

広義にはデータベース利用者のデータに対する見方(データの論理的な構造)。データベースシステムの記憶装置内のデータの扱い(データの物理的な構造)そのものは，すべての人に開示できない，人間にわかりにくいなどの理由で，利用者に見せるための機能や操作法が定めてあり，これをビューということもある。システム固有の定義によって異なり，ネットワーク型データベースでは外部スキーマの一部であり，関係データベースでは，関係演算の結果としての関係がビューである。

ヒューマンインタフェース(human interface)

人間とシステムの境界にある情報の相互伝達の手段。すなわち各種のエレクトロニクス機器と人間の間に存在し，人間がそれを操作することで機械に人間の意思を伝え，またエレクトロニクス機器は表示装置等に応答・結果等を表示すること。人間とシステムの間の対話を行えるような装置で，各種OA機器，VDT機器などがこれに相当する。

エンドユーザに使いやすいコンピュータシステムをめざすうえで，ハードウェア面，ソフトウェア面を含め表示方式や入力方式をくふうするなどして，ヒューマンインタフェースの向上を図っている。ユーザインタフェース，マンマシンインタフェースともいう。

表(table)

①広義には，互いに関係のある項目が，その相対的位置によって識別できるようにデータを配列したもの。→(類)配列

②データや情報の関係や意味を整理し，わかりやすく表現した書面。文字，数字，記号を主体とするもので，図的要素により視覚化するグラフと区別する。一般的には，縦横の升目を利用する。(1)一覧表，(2)分類表，(3)比較表(対照表)，(4)評価表などがある。→(参)グラフ

表意コード(mnemonic code)

①人間の言語を連想しやすく規定したコード。コンピュータが実行できるプログラムは 2進数の列で構成するが，直接 2進数でプログラムを組むのは不便である。そこで，コンピュータの各命令を適当な覚えやすい略号にして人間が扱いやすくしたアセンブラ言語ができた。このように人間の言語を連想しやすく規定したコードを表意コード(擬似コード，ニモニックコード)という。→(類)アセンブラ

②コード化の一手法。コード化対象の属性を連想しやすい文字や数字で表現するコード。略語コード，子音コードなどが代表的である。

日本→JPN，米国→USA，20インチカラーテレビ→TVC-20

表計算ソフト(spreadsheet program／worksheet program／table oriented database)

定義した表にデータ作成をすると同時に，式や関数を入力して縦横の集計処理などを簡便に行うことができる計算ソフトウェア。表は縦横に分割してあり，縦方向を行(row)，横方向を列(column)，それぞれが交差している部分を項目欄またはセル(cell)という。文字や数値などのデータは，画面に表示されたセル単位で入力する。2行4列目のセルをR2C4などと指定する。行を数字，列をA，B，C，……で示し，C2とセルを指定するソフトウェアもある。

表計算ソフトが手書きの集計表と違うのは，このセルの中に数値と数式の両方を入れられることである。しかも数式を入力した場合は，セルにはその結果を表示する。

スプレッドシートの機能を拡張して，表の大きさを自由に変える機能，再計算機能，自動実行機能(マクロ機能)，データベース機能，グラフ作成機能，画面の分割などさまざまな機能を強化した製品が多く，これらを特にパワースプレッドシート(多機能表計算ソフト)という。

→(参)アドインソフト

標準化(standardization)

同じ基準で使用できるように品質，寸法，機能，形状などに共通の規格を定めること。コンピュータの分野では，記憶媒体への記録方式，プログラム言語，データ通信のプロトコル，機器の相互接続，データやコードの表現などの標準化が進んでいる。標準化の国際機関としてISO(国際標準化機構)があり，日本ではJIS(日本工業規格)がさまざまな分野の標準化を進めている。→(類)規格，(参)互換，オープンシステム

標準偏差(standard deviation)

分散の正の平方根。ばらつきの度合を示す値で，測定値と単位が一致している。

$$\text{標準偏差} = \sqrt{\text{分散}}$$

ヒ

標準ライブラリ関数(standard library function)

Cで用意されている基本的なライブラリ関数のこと。ユーザ側が用意するライブラリ関数と特に区別してこのように呼ぶ。

表操作(tabel handling)

COBOLの用語で，OCCURS句によって定義した表を用いて，探索・整列，集計などの処理を行うこと。

標本(sample)

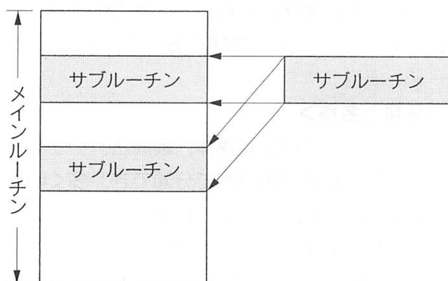
調査の対象(母集団)から，その性質や状態を明らかにする目的で抜き取ったもの。試料，サンプルともいう。

表明(assertion)

推論や計算の実行過程の各時点で，変数間の条件や関係を記述した論理式のこと。特に，ループの中で刻々と変化する変数の値が，常に一定の条件を満たしているときループ不変(loop invariant)といい，アルゴリズムが正しいことを証明する根拠になる。すなわち，問題の答えを与える変数が，プログラム実行中のどの段階でも問題の与えた条件を満たしていなければならない。

開いたサブルーチン(open subroutine)

同一性格の処理内容のルーチン(サブルーチン)をプログラム中の必要な箇所に挿入できるようにしたサブルーチン。一般的にはソースプログラムライブラリに登録しておき，コンパイル，あるいはアセンブルの際に自動的にソースプログラムの中に組み入れる。アセンブラのマクロルーチン，COBOLのCOPYルーチンなどが代表的。



→(関)サブルーチン，(対)閉じたサブルーチン

フ

フ

ファームバンキング(firm banking)

銀行のコンピュータと企業のコンピュータ(端末)を通信回線で接続し，銀行の窓口業務サービスを企業内で受けられるようにしたシステム。

銀行の第3次オンラインシステムにより普及が進んだ。基本的なサービスは，口座入出金，振替，給与振込などであるが，銀行がもつ金融情報や企業情報の提供も行っている。ファームバンキングは，企業にとっては銀行取り引きの結果が瞬時に分かり，資金の効率的な調達・運用ができるほか，経理部門の事務や入力負担の軽減，売掛金の自動消込みなどのメリットがある。いっぽう，銀行にとって，行内事務の合理化のほか，企業との取り引きをシステム化し，強固な取り引き関係を作ることができる。

ファイラ(filer)

補助記憶装置内のファイル操作を容易化，簡易化するユーティリティソフト。ファイルのツリー構造表示，複写，削除，ファイル名変更，ディレクトリ(フォルダ)移動，属性変更などをコマンドを用いず，メニュー操作やアイコン操作で行うことができる。

ファイル(file)

①通常外部記憶装置上に作成するデータやプログラムの集合のこと。ファイルは論理的な概念であり，ファイルが作成される物理装置の性質を反映する。例えば，ディスク上に作成されたファイルはランダムアクセスが可能であり，テープ装置の場合はシーケンシャルアクセスのみである。

②プログラムで処理する関連のあるレコードの集合体。レコードは，ある性質と目的をもったデータの並びであって，利用者が作成する。

→(関)レコード，項目

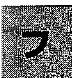
＜ファイルの分類と名称＞

ファイルは，①の視点では記録媒体，編成法，アクセス方法などで，②の視点では記録した内容や使用目的で分類でき，多くの種類名称がある。

(1)用途での分類：マスタファイル，トランザクションファイル，ワークファイル，経歴ファイル，バックアップファイルなど

(2)使用期間での分類：パーマネントファイル，テンポラリファイルなど

(3)編成法での分類：順次編成ファイル，索引順次編成ファイル，直接編成ファイル，区分編成ファイル，VSAM編成ファイルなど



(4)ボリュームでの分類：単一ボリュームファイル，マルチボリュームファイルなど

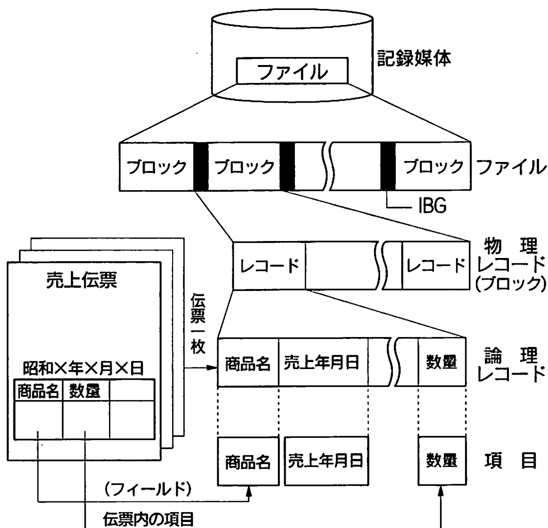
(5)アクセス法での分類：直接アクセスファイル，順次アクセスファイル

(6)記録形式での分類：バイナリファイル，テキストファイル

(7)記録内容での分類：データファイル，画像ファイルなど

(8)利用主体での分類：システムファイル，ユーザファイル，共有ファイル，個人ファイルなど

この他，ファイル処理の場では，入力ファイル，出力ファイル，転送ファイル，参照ファイルなど，適当な名称が使用される。



ファイル管理(file management／library management)

重要なソフトウェア資源であるプログラムやデータを，ファイルという形式で集中的，体系的に管理すること。また，これらをプログラムのライブラリ，ファイルのライブラリといい，これらの管理機能全般をさす。ライブラリは保守や再利用のために保存されるもので，適切に管理されなくてはならない。次の5つの視点で行う。

(1)スペース管理：記憶媒体の有効利用，使用頻度に応じた配置などの目的で，記憶領域の再編成，再構成を行う。

(2)共用管理：複数のユーザ，複数のプログラムが参照したり，更新するファイルの整合性を維持する。

(3)障害管理：万一の障害に備え，ファイルの二重化(ミラーリング)やバックアップを行う。

(4)変更管理(世代管理)：更新や変更のたびに変更内容を記録，バージョン管理を行う。変更前のファイルも規定した世代分を保存する。

(5)統計管理：ファイル管理のための統計を作成する。

ファイル管理システム(file management system)

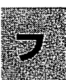
多数のファイルを管理するためにOSが備えている機能。MS-DOSやUNIX系OSでは，ファイルを木構造のディレクトリによって階層的に管理する。WindowsやMacOSではこれらをフォルダと呼ぶ。ディレクトリは，記憶媒

フ

体が記憶したファイルの保存・管理のためにファイル名や大きさ，作成日，記憶位置，参照モードなどの情報を記憶している。このファイル管理方式を木に例えると，ファイルの位置は葉に，ディレクトリは小枝，大枝に相当する。根に相当するディレクトリをルートディレクトリという。
→(関)ディレクトリ

ファイルサーバ(file server)

大容量の磁気ディスクや光ディスクなどの外部記憶装置をもち，ネットワークの利用者(クライアント)が自分用のファイルや共用ファイルを効率的に作成，利用できるようにしたコンピュータ。ディスクサーバよりファイルの排他制御などの管理がいきとどいている。また，多数のクライアントから同時にアクセス要求を受けるため，エレベータ・シーキング(アクセス要求の順番に磁気ディスクをアクセスするのではなく，現在のアクセスアームの進行方向で近くに存在する要求アドレスからアクセスする方式)など，性能を高くするためのくふうを行っている。
→(類)データベースサーバ



ファイル設計(file design)

データの格納や参照に関する方法や構造に関わる設計。データの種類，量，用途，参照・更新の頻度などを考慮して，ファイルの構成(データベース構造，ファイル編成など)，様式を決定する。

ファイル属性(file attribution)

ファイル操作の誤りからデータを保護するためにユーザやシステムがファイルに与える機能上の性格。例えば，読出しは許すが書込みは許さない，アクセスできるユーザを制限する，アクセスの記録をとる・とらない，ファイルの存在をディスプレイに表示しない，バックアップを常時とる，あるいは複写を禁じる，などのファイルに付与された性質のこと。MS-DOSでは4つの属性を与えることができる。
(1)アーカイブ属性：すべてのファイル操作コマンドの対象となる。
(2)隠し属性：ファイル名を表示できないようにする。
(3)システム属性：OSのみが使用できるファイルとする。
(4)リードオンリー属性：削除，更新ができないファイルとする

ファイルダンプ(file dump)

デバッギングの際に，必要により，ファイルの内容を出力すること。

ファイルの更新(update)

マスタファイルのレコードをトランザクションファイルのレコードによって，追加・削除・変更し，最新の状態に保つこと。アップデートともいう。

ファイルの照合(matching)

突合せ，あるいは，マッチングともいう。二つ以上のファイルについて，レコードの構成要素を相互に比較することをいう。一般に照合の結果，必要なデータの抽出・編集・マスタファイルの更新などを行う。

ファイルの処理(file processing)

ファイルに関わる処理のこと。よく行われる処理とポイントは次のようなものである。

(1)単一ファイル処理(ただ一つのファイルに対して行う処理)

a：入力媒体変換と入力レコードの検査

入力されたデータを本格的な処理に先だって，内容をチェックしたり，書式を整えたりする処理。

b：帳票印刷処理(見出し処理，改ページ処理など)

ファイルの内容を読みやすい形式にプリントしたり，集計したりする処理。

c：グループ制御処理(コントロールブレイクなど)

レコードのキー値によって，順番を検査したり，グループ集計したり，特定の値以上のレコードを抽出する処理。

(2)複数ファイル処理(複数のファイルについて行う処理)

a：突合せ処理(マッチング処理)

二つ以上のファイルの論理関係を確認しながら行う処理。関係によって処理内容が分岐する点に特徴がある。

b：併合処理(マージ)

一定の順序で分類してある(ソートずみの)二つ以上のファイルを，一つのファイルに同じルールでまとめあげる処理。

ファイルの分類(sorting)

指定された基準にしたがって，レコードやデータをグループに分けたり，一定の順序(降順・昇順)に並べ変えること。→(同)ソート

ファイル媒体(file medium)

ファイルを記録・保存できる媒体。磁気テープ，磁気ディスク，光ディスクなどの外部記憶媒体をいう。

ファイル編成(file organization)

磁気テープ記憶装置，磁気ディスク記憶装置などの外部記憶装置の機能とアクセス方法を適合させるファイルの論理的な構成，編成法の総称。具体的には下記のような編成，VSAM編成がある。

編成法	外部記憶装置の種類	アクセス方式
順次編成	MT・DASD	順次
直接編成	DASD	順次・直接
索引順次編成	DASD	順次・直接
区分編成	DASD	順次(直接)
相対編成	DASD	順次・直接

ファイル名(file name)

多数のファイルを識別，特定するためにそれぞれのファイルに開発者やユーザが付ける名称(拡張子を含む)。ファイル名に使用できる文字や長さはファイル管理システムにより異なる。階層構造のファイル管理システムでは，ディレクトリが異なれば，同じファイル名でも別のファイルとして扱う。MS-DOSでは，ASCIIコードで8文字＋拡張子3文字の制約があったが，Windowsではより長いファイル名を許容するようになった。

ファイルラベル(file label)

ボリューム内に存在するファイルの情報が記述された内部ラベル。磁気ディスクでは，VTOCにすべてのファイルのファイルラベルが集中管理されている。その内容は(1)ファイル名，(2)作成年月日，(3)保存期間，(4)ファイル編成法，(5)レコード形式，(6)エクステント情報(ファイルの記憶領域)である。

また，磁気テープのファイルラベルは，見出しラベルと後書きラベルから成り，それぞれのファイルの先頭レコードの直前と最後のレコードの直後にある。

ファイル利用権の管理(file access permission)

利用者または作業工程(部門メンバ)に対して与えたファイルのアクセス権を管理すること。情報の完全性保持，機密保護のための常套的な対策である。ファイル利用権には，読取り許可，書込み許可，読取り書込み許可があり，利用者のIDコードとパスワードを登録し，アクセス権の認証には双方の合致が条件となる。なお，IDコードとパスワードは別々に管理し，遺漏があってはならい。→(参)ファイルの属性

ファイルレイアウト(file layout)

ファイル設計作業の工程で，データをファイルに格納する際の，項目，桁数，属性などを決定すること。→(類)ファイル設計

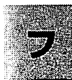
ファクトデータベース(fact database／source database)

検索の目的となる情報(ソース情報)そのものを提供するデータベース。コンピュータの処理スピードや記憶容量などの限界が拡大するにつれ，フ

アクトデータベースが誕生した。リファレンスデータベースでは情報の所在を示すだけであるが，情報そのものの提供ができるようになったことにより，商用データベースサービスにファクトデータベースが増加した。今日ではデータベース利用者の大半がファクトデータベースを対象にしている。→(類)フルテキストデータベース，(対)リファレンスデータベース

ファクトリオートメーション(factory automation：FA)

製造工場で，製品開発から製造工程まで情報化と自動化を進め生産性と柔軟性の向上を図ること。CAD／CAMにより製品の設計段階からコンピュータを使い，製造工程でのオートメーション化まで含む。工場では，多くの設備，装置，機械，材料を利用しており，連携，統合化の技術はさまざまである。製造段階ではNC工作機械，産業用ロボットなどにより，部品の運搬から組立，検査，梱包，出荷に至る全工程での自動化，省力化が進み，多品種少量生産などの社会的ニーズに対応できる生産体制づくりを目指した統合的オートメーション化が進んでいる。→(参)CAD，CAM，NC工作機械，産業用ロボット，CAE，CIM



ファジィ(fuzzy)

人間の判断がもっているあいまいさ。通常のコンピュータ処理では厳密さを要求するが，このあいまいさを積極的に応用し，処理する学問がファジィ工学である。人工知能(AI)の研究の一分野で，処理を次のステップへ進めるときの判断条件に，人間のカンのようなあいまいさを加味してプログラム化している。ファジィの考え方による制御方式は，一部の地下鉄の運行，家庭電化製品の制御などに実用化されている。

フィールド(field)

1件のレコードを構成する要素で，レコードのもつ各データの格納領域のこと。表計算ソフトでは縦に並ぶ領域であり，レコードの構成によっては1レコード中に複数個のフィールドがある。アイテム(項目)ともいう。

例えば，社員レコードの中の社員番号欄，社員名欄，住所欄などの欄をさす。→(同)項目，(参)領域

フィールドテスト(field test)

テストの最終段階で，実際に運用するコンピュータシステム，ネットワークシステムを使用し，実際のデータを用いてテストすること。

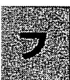
フェールセーフ(fail safe)

コンピュータシステムに故障や不良が生じたときに，あらかじめ設定したより安全な状態のほうへシステムをダウンさせること。障害が拡大しない，被害を大きくしない，誤動作の連鎖反応でシステム全体に故障を招か

ない，危険な出力を出さないなどの目的で意図的にシステムを停止させ，他の資源やデータを保護すること。コンピュータシステムは，障害を想定して，障害発生の場合でも発生以前と同様の処理能力を維持し処理を続行できるように安全を考慮して設計する。フォールトトレラントコンピュータが後者の例。→(類)フェールソフト

フェールソフト(fail soft)

コンピュータシステムに故障や不良が生じた時，障害箇所を放棄したり，被害を最小限に抑えながら，ある程度の仕事を続行すること。システム全体の動きを停止するのではなく，正しく使える限りの物を使って，システムの処理性能は低下しても，その機能をできる限り維持する手法。システムには，適切なバックアップが必要である。例えば，専用回線が使用できなくなった場合，即座に公衆回線に切り換えて，低速ではあっても処理を続行することなどが相当する。→(類)フェールセーフ



フォーマット(format)

→(同)書式

フォルダ(folder)

GUI環境の階層的ファイルシステム(HFS)のディレクトリに相当する概念。名前をもった書類フォルダのアイコン1個で示すファイルの集合。サブディレクトリと同様，フォルダの中にフォルダを入れることも可能である。フォルダをクリックするとその中身をウィンドウに表示する。フォルダにファイルを入れる操作は，ファイルのアイコンをドラッグして，フォルダのアイコンに重ねればよい。→(類)ディレクトリ

フォルトトレランス(fault tolerance)

システムの信頼性を高めるための機能。システムに冗長性を導入して実現することが一般的である。耐故障技術ともいう。ハードウェアの障害に対処するには余分な回路・部品・装置を付け加える。ソフトウェア障害には，プログラム誤りの収集分析，故障診断プログラムの利用，二重処理，照合などを行う。耐障害機能を高め，常時稼働能力をもつシステムを，特にノンストップコンピュータという。→(参)デュアルシステム

深さ(depth)

木構造の表現で，ある頂点とルートにある頂点との隔たり。2頂点間の枝の数に相当する。最大の深さ(ルートから最も遠い頂点の深さ)を特に木の高さ(height)という。→(対)幅

深さ優先探索(depth first search)

木構造のグラフのデータ探索法の一つで，ルートからスタートし，分岐点

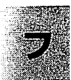
では常に一つの経路を選択しながら，次第に深い頂点へとたどる探索法。葉に至れば，経路を逆に引き返し前の頂点に戻り，そこからまだ探索していない経路があれば探索を続行する。探索に成功したときは，経路が記憶されている。失敗したときにはルートに戻ってしまう。→(対)幅優先探索

負荷テスト(load test)

一般にシステム開発時のテストは，少量のデータ，回線，端末で実施されることが普通で，本稼働の際，回線や端末から大量のデータが投入された場合のテストがない。この点を特にテストすることを負荷テスト，または過負荷テスト(設計目標以上の負荷をかける)という。端末がいっせいに処理を要求したとき，レスポンスが目標値を満たすかなどを検査する。

復元(reconstitution／recovery)

システム運転中に，ハードウェアやソフトウェアの障害によって処理が異常終了した場合，正常な処理の終了時点までさかのぼって，プログラムやデータを復旧させること。通常，磁気テープなどに記録しておいたバックアップファイルを用いて，バックアップを取得した時点まで戻し，ログを見て，それ以降の処理を再度実行して正常な状態にする。



副作用(side effect)

①デバッギングの目的で修正したあるプログラム部分が他のプログラム部分に悪影響を及ぼすこと。このためプログラムを修正した後は，結合する他のモジュールを含めて相互影響をテストし直す必要がある。

②操作に伴って，2次的に実行環境の変化が起こること。Cでは，例えば，volatile(揮発性オブジェクト)の操作，ファイルの変更等で副作用が引き起こされる。また，式の評価でも副作用が発生する場合がある。

副スキーマ(subschema)

→(同)サブスキーマ

副プログラム(subprogram)

広義には，モジュール化したプログラムで，上位のモジュールから見て，下位のモジュール(submodule)を副プログラムという。例えば，メインプログラムから呼び出して利用するプログラム。代表的なものにサブルーチン副プログラムがある。このほか関数副プログラム，初期値設定副プログラムなどがある。

プッシュダウン(push down)

スタックに新しいデータを記憶させること。単にプッシュ(押し込み)ともいう。→(参)スタック

物理データ設計

論理データ設計(ファイル設計)で定義されたファイルのデータ項目・データ量や処理内容から，ファイルの編成方式やレコードレイアウトを定義する工程。ハードウェア(記憶装置や記憶媒体)を意識したファイル設計で，データベース構造もこの時点で定義される。通常，次のような順で行う。

(1)マスタファイルと関連ファイルのデータ項目，編成法の決定

- ファイルの特性，管理される項目の把握，他ファイルとの関連
- ファイル構造と編成法の決定
- 記憶媒体の決定，容量，入出力時間の算出

(2)レコードレイアウトの設計

- 欄，項目，フィールド，レコード，ファイルの属性，サイズ
- 固定長レコードか可変長レコードか
- レコードレイアウトの決定

物理レコード(physical record)

磁気テープや磁気ディスクなどに対して，レコードを入出力するときの単位。ブロックともいう。複数の論理レコードが集まって，一つの物理レコードが構成される。→(対)論理レコード，(関)ブロック化

不定長レコード(undefined length record)

可変長レコードと同様に論理レコード長がそれぞれ異なるが，おのおのの論理レコードが自分自身のレコード長情報をもたない形式。レコードの入出力はIBGをもとに行うため，ブロッキングはできない。
→(関)可変長レコード，(対)固定長レコード

部品化(divide／division)

部品とは，標準化されたソフトウェアを，再利用する際の再利用の単位のこと。従来の共通モジュールの考えを拡げたもので，再利用を前提にした標準化により，他のソフトウェアの構成部品になるようにモジュールを作る。これらを整理，分類のうえ部品ライブラリの形で保管しておき，必要に応じて再利用するものである。部品モジュールの数は大量になるため，登録，検索，結合などの作業を支援するツールの活用が不可欠である。

部分順序木(partially ordered tree)

2分木の一つで，親(ルートに近い頂点)のデータが子(親から出た枝の先の頂点)のデータより小さくならない木のこと。2分探索木が枝の左右でデータの大小を割り振ったのに対比して，部分順序木は枝の高さ(深さ)で大小を割り振る。ソートには適さないが最大値探索に適している。優先度の順位のある待ち行列の実現などに利用する。

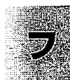
プライオリティ(priority)

→(同)優先順位

プライバシ保護(privacy protection)

コンピュータセキュリティに関連して，個人に属する情報や個人に関する情報の利用・流出を制限し，個人の権利(privacy)を守ること。積極的には，自分に関する情報は自分で管理できる権利を含む。例えば，銀行取引，クレジット利用状況，病歴，犯罪歴など，個人の人権，信用，秘密にかかわる情報を調査機関などが勝手に収集できないようにアクセスを制限すること。また自分に関する情報の開示を求め，訂正できるようにすることで情報の悪用を防止し，人権を保護する。アメリカでは1980年にプライバシ保護法が成立している。日本では1988年に個人情報保護法を公布した。

フラグ(flag)



標，標識のこと。目印，マーク，旗ともいう。

①ハードウェアやデータなどが特定の状態にあるかどうかを表すビット，バイト，キャラクタ，語。分岐命令が飛越し先を判断するために評価する。フラグの役割を果たす変数はプログラム上で用意する場合と，条件コードレジスタのようにハード的に用意する場合がある。フラグが2値のいずれかしかとらない場合，特にスイッチということもある。

②ファイル，レコード，ブロックなどのデータの集合の始め，または終わりを示すための符号。

フラグメンテーション(fragmentation)

ファイル装置上の空間を区分しておき，必要な大きさだけファイル空間を割り当てる方式(例えば区分編成ファイル)では，ファイルの更新やレコードの削除により，空き空間(無効領域)ができる。この空き空間をさす。

ブラックボックステスト(black box test)

プログラムのテスト法のひとつで，テストデータの選定・作成する際に，プログラムの内部構造・制御の流れにかかわらず，その外部仕様に基づいて考えられるデータを選ぶ方式。つまり入力される可能性のあるさまざまな組合せのデータを想定し，プログラムのすべての機能をテストするねらいをもっている。機能テスト法ともいう。結合テスト以降の段階で行うことが多い。テストデータの作成には次のような方法がある。

(1)同値分割：入力データの範囲を有限のクラスに分割し，各クラスの代表値をテストデータとする。そのクラスの代表値でテストすれば，クラス内のデータをすべて検査したことになるという考え方。

(2)限界値分析：入力データと出力データを同値クラスに分割，各クラス

の境界値の両端をテストデータとする。

(3)原因-結果グラフ：プログラムの外部仕様から，入力値，条件(原因)の差で，出力の値(結果)がどう変わるかの関係を調べ，グラフにし，グラフからテストデータを作成する。

(4)エラー推測：ありそうなエラーを推測し，検出するためのデータを作る。推測に根拠がない限り有効とはいい難いが，初歩的な論理エラーやチェックのあまさは発見できる。→(対)ホワイトボックステスト

プラットフォーム(platform)

もともとは人または物を載せる台の意味。工学では，あるシステムを搭載するための別のシステム。情報処理では，あるソフトウェアやコンピュータシステムを動作させるための基盤となるハードウェアおよびソフトウェア環境，端的には動作環境をさす。例えば，「最近の電子メールシステムの製品には，UNIXをプラットフォームにしたものが多い」というような使い方をする。



フリーウェア(freeware)

フリーソフトの別称。

フリーターム(free term)

データベース検索の際に用いる語で，タイトル，コンテンツにその語が存在すればすべて検索の対象とする語。文献や論文のデータベースでは登録時にその内容を表すいくつかのキーワードを登録してあるため，キーワードの組合せで効率よく目的の対象を検索できる。しかし，適当なキーワードがない場合などキーワードに代えて任意の自然語(フリーターム)で検索でき，主題(タイトル)と本文(コンテンツ・テキスト)を検索する。

プリコンパイラ(pre-compiler)

ソースプログラムのコンパイル前に用いる言語処理プログラム。あるプログラム言語の機能を拡張し，その拡張した表現を用いて書いたソースプログラムを標準的なソースプログラムの形式に変換する処理を行う。その後は通常のコンパイラで翻訳し，オブジェクトプログラムにする。

プリプロセッサ(preprocessor)

①プリプロセス処理(前処理)を行うプログラムのこと。トランスレータの一種。前処理命令の解釈，ファイルのinclude，文字の置換，マクロ展開等の処理を行う。出力はコンパイルの単位(翻訳単位)となる。

②特定のプログラムのため，入力や処理をしやすくするような形にデータ形式や計算手順を前もって変換する機能をもつプログラム。

プリントサーバ(print server)

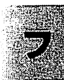
クライアントサーバシステムの中で，プリンタをもちネットワーク利用者(クライアント)からのプリント要求を専門に処理するコンピュータ。大容量の磁気ディスクと高性能のプリンタを装備し，スプール機能を使って効率的に印刷する。

フルテキストデータベース(full text database)

論文や新聞記事などの全文データベース。文献・論文データベースに登録してあるデータを特定する表題や見出し，著者名，収録ライブラリ，要約，キーワードなどの部分をタイトル(title)といい，対象となる論文や記事の本体をコンテント(content)，またはテキスト(text)という。

→(類)ファクトデータベース，(対)リファレンスデータベース

ブレイクポイント(break point)



①コントロールブレイク処理で，処理手順を変えるための区切り点。例えば，在庫の商品別集計の，商品コードの変わり目など。

→(関)コントロールブレイク，グループトータル

②プログラムのデバッグや実行状態の監視のため，プログラムの実行を中断し，外部から割込みがかけられるように，プログラム内に指定した場所。チェックポイント(check point)，区切り点ともいう。プログラム内にブレイクポイントを設定することで，プログラムの実行過程を追うこと，実行中のある場所における記憶領域のダンプをとること，デバッガの呼出しを行うことなどの処理を，ハードウェアの助けを借りずに行うことができる。

→(関)デバッグ

ブレーンストーミング(brain storming)

新規のアイデア，実行可能なアイデアを探し出すために行う集団による会議。原義は「突然の思いつき」の意。通例2チームを設ける。

(1)アイデアを出すチームでは，自由な着想と発言を制約しないために，他人のアイデアを批判しない，他人のアイデアに自分のアイデアを付加する，できるだけ多く発言するなどのルールを設ける。狭義にはこのアイデア会議をさす。

(2)こうして提案された奔放なアイデアを整理，評価し，具体案をつくるチーム。

フローチャート(flowchart)

問題の定義，分析または解法の図的表現。流れ図。演算データ，流れ，装置などを表現する流れ図記号を組み合わせて手順やプロセスを表した図

表。1947年，H.Goldstineとvon Neumannが提唱したといわれている。流れ図を用いることによって作業やプログラミングの流れが正確に把握でき，第三者にも正確に伝えることが可能となる。流れ図はオペレーション，判断，コネクタなど，一定の形式，順序，要素にしたがって書く。流れ図にはプログラムフローチャートとシステムフローチャートがある。プログラム流れ図は，コンピュータでの処理手順を概括的に図式化してまとめあげたもので，コーディングのもとになる。システム流れ図は，データが発生してから処理を行って情報を得るまでに，システムの各部分のデータが流れるさまを図式化したものである。システム流れ図を作成することにより，データの流れとその記録媒体，利用する機械と操作などの関係を明確にできる。

プログラマ(programmer)

プログラミングをする人。問題処理プログラムを作成するアプリケーションプログラマと，ハードウェアやオペレーティングシステムに関連したプログラムを作成するシステムプログラマとに分かれる。

プログラミング(programming)

広義には，プログラムを設計し，書き，テストすること。プログラム仕様書に従ってデータ構造とプログラムの制御構造を決め，プログラム言語で記述し，デバッグし，テストを行う工程。狭義には，プログラム言語により，プログラムをコーディングすること。

プログラミング環境(programming environment)

システムエンジニアやプログラマに提供(サービス)するハードウェア，ソフトウェア両面の開発ツール，装備やその整合性の状況。プログラム開発のためには多くの工程を経るが，各工程でシステムエンジニアやプログラマが利用するソフトウェア(ツール，サービスプログラム)も数多い。これらのツールが揃っておりツール間で情報を有機的に授受できると，ソフトウェアを効率的に作成できる。

プログラミングシステム(programming system)

プログラムの作成を容易かつ効率的にするために使うソフトウェアやその利用法。プログラム言語，エディタ，言語処理プログラム，ユーティリティプログラムなど。→(参)開発システム

プログラム(program)

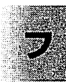
コンピュータに処理させる作業を指令するための手順をプログラム言語により記述した文。一連の命令や文の集まり。その内容や処理の目的によりさまざまに分類できる。狭義にはユーザがプログラム言語を用いて記述

するアプリケーションプログラムをさす。広義にはオペレーティングシステムやハードウェア組込みのシステムプログラムを含む。
→(関)オペレーティングシステム，(類)ソフトウェア

プログラム開発(program development)

広義には，システム開発工程におけるプログラム設計からテストまでの広い段階。狭義には，プログラム設計の仕様書に基づいて，各種プログラム言語によりプログラムを記述し，デバッグ，コンパイル，テストなどを経て運用可能なプログラムを作る工程。主としてプログラマの仕事である。プログラム開発の効率化をめざし，デバッグツール，CASEツールなどの開発支援ツールの利用も進んでいる。
→(参)システム開発，(参)システム設計，(参)CASE

プログラム間インタフェース(interface between programs)



プログラム相互間のインタフェース。モジュール間インタフェースと同義。プログラム間で受け渡すデータを明確にする工程を，プログラム間インタフェースの決定(あるいは定義)という。これは，内部設計の機能分割・構造化の段階で行い，一般にデータフローダイヤグラム(DFD)やデータ流れ図で表現する。

プログラム構造化設計(structured design of program)

プログラムを小さなモジュールの集合体として構成するプログラム設計の手法。次のような手順が一般的である。(1) 最上位モジュールの定義，(2) モジュールの機能分割，(3) 最適な分割技法の選択，(4) モジュール分割，(5) モジュール間インタフェースの定義，(6) モジュール分割の妥当性の評価と再分割

プログラム情報(information for program design)

プログラムの内部設計に必要な設計情報の一つで，主としてプログラムの仕様設計に必要な情報の集合。プログラム関連図，プログラム構造図，モジュール構造図，プログラムで使用するファイル一覧，共通テーブルを使用するプログラムの一覧，プログラム間の相互参照表などが含まれる。

プログラム設計(program design)

システム設計(内部設計)の仕様に基づいてプログラムを開発するための設計段階。プログラム言語，機能別のモジュール分割，プログラム間のインタフェースなどを定め，プログラム仕様書としてまとめる。

プログラム流れ図(program flowchart)

プログラム設計の段階で，コーディングの指針を与えるためにプログラムの処理手順を概括的に図式化したもの。流れ図は，データを処理して情

報を得るまでに必要なデータの流れとその処理，記録媒体，利用する機械と操作などの関係を明確にする。

プロジェクトチーム(project team)

会社組織上の部門ではなく，特定の目標を実現するために構成されたチーム。システム開発では，開発部門が主体となるが，ユーザ部門(顧客の場合もある)の開発推進組織・技術支援グループなどの外部組織と連係したチームで，システムが完成すれば使命を終わる。システム開発の各局面において，必要な人材の能力や人数が異なるため編成や役割の変更が多い場合もある。

プロジェクトマネージャ(project manager)

情報システム開発のプロジェクト管理にかかわり，システム化計画に基づきプロジェクト計画を作成し，プロジェクト体制の確立，必要資源の調達ができ，またプロジェクトの進捗・品質・要員等の管理を行う技術者。次のような役割を担う。

(1)人材，資源を活用し，予定の費用・工期・品質でシステムを完成する。

(2)外部組織と適正な契約を結び，円満な関係を保持する。

(3)ソフトウェア生産技術スペシャリストと協力して総合的な品質保証を実現する。

プロセス(process)

処理を意味する用語。計算(computing)，アセンブル，コンパイル，編集，更新などの意味を広く含む。

プロセス制御(process control)

①工業プロセスで，温度，圧力，流量などの物理量をリアルタイムで制御すること。センサなどを通じ装置を監視し，コンピュータでシステムを制御するには厳しい処理能力が必要。

②オペレーティングシステムではタスク制御と同義。

プロセス制御システム(process control system)

化学・石油・ガス・製紙・鉄鋼などの工場(プラント)の連続的な生産工程(プロセス)で，圧力・温度・液面・濃度などの物理的な状態量を制御する目的で構成された系。例えば，製造物を一定の環境で生成するとき，制御対象の状態を検出器で検出し，比較部で基準入力信号と比較し，その結果(制御動作信号)を調節部で操作信号に変換し，操作部に伝えるシステム。外界の影響で環境に変動があっても，この操作で目標とする環境を維持する。微小な環境変化が製品の品質や安全性を左右する系では，リアルタイム制御処理の技術が適用される。

プロセスフローチャート(process flowchart)

作業工程の順序を，流れ図記号を用いて見やすく表現した図表。

プロダクションエンジニア(production engineer)

情報システム開発にかかわり，内部設計，プログラム設計，プログラム作成，テスト，移行を行う技術者。次のような業務を行う。

(1)アプリケーションエンジニアの作成した外部設計書に基づきシステムの内部構造とその仕様を定義する内部設計

(2)内部設計から，標準化，品質，保守性等を考慮したプログラム作成

(3)テスト計画の作成，実施。テスト結果の分析と信頼度予測

(4)保守計画の作成，実施

ブロッキング(blocking)

複数の論理レコードをひとつの物理レコードにまとめること。ブロック化ともいう。ブロッキングの目的は，

(1)中央処理装置と外部記憶装置の間でファイルの読み書きを行う場合，入出力回数を減らすことでの処理効率の向上

(2)IBGの削減による記憶領域の有効利用

ブロッキングの指示はJCL(ジョブ制御言語)で行うため，ユーザプログラムの中で特に意識する必要はない。→(関)ブロック，(対)デブロッキング，(参)論理レコード，物理レコード



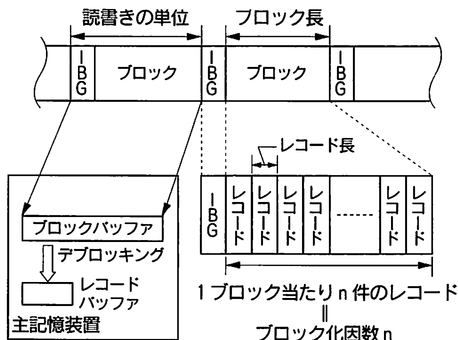
ブロッキングファクタ(blocking factor)

→(同)ブロック化因数

ブロック(block)

一般にひとかたまりのデータまたはプログラム。

①磁気テープ，磁気ディスクなどの補助記憶装置に情報を記憶させる場合，読み出す場合の，転送単位となる一連の文字や語の集まり。1物理レコードと同じ。ブロックの長さ(文字数，バイト数などで表す)をブロック長(block length：BL)という。



②プログラムの中で共通の名前(変数名など)を使用できる範囲。サブルーチンや副プログラムが，メインルーチンやメインプログラムの指定した名前とは別の名前を利用できるとき，それぞれの名前の有効範囲がブロックである。

③構造化されたプログラム中でひとまとまりの処理をする範囲。→$$

④データ通信の分野では，通信システムの規約や能力に合わせ，長大な通信データを適当な長さの複数個のデータのかたまりに分割した単位。

ブロック化(blocking)

→(同)ブロッキング

ブロック化因数(blocking factor)

1ブロック(1物理レコード)にまとめた論理レコードの数。ブロック化係数ともいう。ブロック化因数は，磁気記憶媒体に記録できるレコード数を決定する要素である。ブロック化因数を大きくすれば，

(1)磁気記憶媒体の見かけ上の記録密度の向上

(2)読出しまたは書込みの回数の減少による入出力のスピードアップ

などの利点がある。しかし，内部記憶装置に大容量のバッファを必要とするため限度がある。通常，ブロック化因数の決定は両方のかねあいで行う。

ブロック化係数(blocking factor)

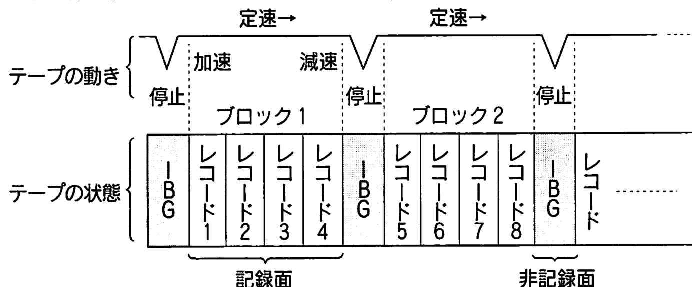
→(同)ブロック化因数

ブロック化レコード(blocked record)

ブロッキングされた物理レコードをブロック化レコードという。また，ブロッキングをしない物理レコードを非ブロック化レコード(unblocked record)という。非ブロック化レコードでは論理レコードは物理レコードと一致する。

ブロック間隔(interblock gap：IBG／block gap)

磁気テープにより多くのデータを記憶させるために，レコードをいくつかまとめてブロック化する。ブロックとブロックの間には何も記録しない空きが生じ，この空きをIBG(ブロック間隔)という。IBGは，10mmから15mmの長さをもつ非記録面であり，読出し書込みの際にテープ走行は，減速・停止・加速の部分に相当する。定速になるとブロックの読出し書込みを行う。ブロック化係数が1の場合，特にIRGという。



上の例では，固定長レコードとし，
ブロック化因数は4である

ブロック長(block length：BL)

ブロックの長さ。文字数，バイト数などで表す。→(参)ブロック

プロテクト(protect)

①コピープロテクト：フロッピーディスク等の記憶媒体からのソフトウェアやデータの無断コピーを防止するために媒体に行う処理。ガードともいう。これを破ることをアンロック(unlock)という。→(同)コピープロテクト

②アクセスプロテクト：書き換えてはならないデータを他のユーザ，プログラムからの更新要求から保護すること。

③シークレットプロテクト：機密のデータを不正アクセスから保護すること。アクセスプロテクトのほかデータの暗号化などを施す。これを破ることをブレーク(break)という。

④データプロテクト：ハードウェアの障害からデータを保護すること。物理ファイルに対する更新の履歴を記録するジャーナリング(journaling)，物理的にデータを二重に書き込むミラーリング(mirroring)などがある。

プロトタイピング(prototyping／prototyping model)

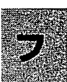
ソフトウェア開発においてユーザの要求仕様が明らかになった段階で，ある程度のプログラムを作成し，ソフトウェアの利用イメージをユーザに

フ

与え，運用してもらった結果からヒューマンインタフェース部分の要求仕様を固めること。またその目的のために作るソフトウェア。開発初期段階でユーザの要求仕様が固まれば，最終段階にはいってからの仕様変更による再設計が少なくなり，手戻りの工程を抑制できる。最近では4GLを用いたプロトタイピングが容易になり，また完成品に組み込むことも可能になった。

フロントエンドプロセッサ(front end processor：FEP)

パソコンなどのシステムに，日本語入力を効率的に行うための日本語入力フロントエンドプロセッサの略。日本語ワープロソフトのかな漢字変換機能だけを独立させたもので，メモリに常駐させて使用し，アプリケーションソフト上での日本語入力を可能にする。



文(statement／sentence)

プログラムの主要な構成単位。実行や宣言の単位となるもので，文法に従っていくつかの命令をつないで終止符と空白でとめたもの。例えば，COBOLでは，文や句を用いてプログラムが書かれる。FORTRANでは実行文(代入文.制御文，入出力文)と非実行文(宣言文，DATA文，FORMAT文，文関数定義文，副プログラム文)を用いてプログラムが書かれる。

分割統治(divide and conquer／divide and rule)

問題を部分に分割して解き，それぞれの部分の結果を組み合わせて全体の解を得ること。例えば，複数個の数値を幾つかの部分に分割して，各部分の最小値を求め，その中から最小値を選べば全体の最小値が選べる。また，例えば，n！(nの階乗)は，n＊(n－1)！で求めることができ，(n－1)！は，さらに(n－1)＊(n－2)！で求めることができるように，問題を次第に小部分に閉じ込めることなどが含まれる。分割統治を行う問題では，分割した部分に共通のアルゴリズムが適用できることが大切で，2分探索，マージソート，再帰プログラムなどに応用できる技法である。

分散(variance)

標準偏差を二乗したもので，データのばらつきを計る尺度。

X_iをデータ，x～を平均値，nをデータ数とすると，

分散$S^2=\Sigma(X_i-x\sim)^2/n$

→(関)標準偏差

分散型データベースシステム(distributed database system)

データを複数のサーバ上に置いて，全体としてのデータベースを管理するシステム。データの配置は，データを分割して複数のサーバに配置する場合と同じデータを複数のサーバに配置する場合がある。前者の利点は，

クライアント(利用者)間で必要とするデータに偏りがある場合，それぞれ別のサーバ上にあればサーバの負荷分散が可能になる。また，後者の利点はサーバの一台に障害が発生しても，他のサーバに同一データがあるため障害の影響を極小化できることである。さらに，サーバ負荷分散も可能である。ただし，後者はデータの同期の維持，データの更新，性能の低下防止などのためのくふうや余分なリソースが必要となる。

→(関)分散トランザクション処理

分散データ処理(distributed data processing：DDP)

マイクロプロセッサをもった多数のインテリジェント端末を回線で接続し，それぞれの端末が互いにデータ処理の役割を分担しながら，全体としてある目的を果たす方式。中央への処理負荷の一極集中を避け，機能を分散して利用現場に密着した処理を行う。これにより端末の特性を生かした柔軟性のあるシステムを構築できる。分散型アプリケーションとして代表的なものに，分散型データベース，オンライントランザクション処理，電子メール，電子会議，グループウェアなどがある。

分散データ処理システム(distributed data processing system)

複数のインテリジェント端末やサーバを分散配置し，互いにネットワークで結合した分散データ処理を行うシステム。垂直分散モデル，クライアントサーバモデル，ピアツーピアモデルなどのいくつかのモデルがある。分散処理システムにすると，処理の一極集中が回避でき，負荷が分散されることで，信頼性が高く柔軟性のあるシステムを構築できる。

分散データベース(distributed database)

物理的あるいは論理的に分散しているデータベース(DB)をネットワークを用いて結合し，どのコンピュータからもひとつのデータベースであるかのようにアクセスすることができるようにしたデータベース。分散DBの機能は，リモートデータベースアクセス(RDA)機能を基盤として，

(1)分散しているDBをひとつに見せる分散透過機能

(2)分散DB定義の管理機能

(3)最適アクセス方法の情報を提供する問い合わせ機能

(4)分散トランザクション管理機能

(5)2相コミットメント制御を中心とした分散同時実行制御機能

(6)障害回復機能

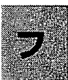
(7)分散DB管理者機能

などが必要である。このような機能によりDBのデータ完全性を保証し(論理的に集中)，かつ実データの存在場所は分散という構成を実現できる。分

散型データベースは機能的に高度で複雑であるが，データが単一のコンピュータに集中しないため，多数のユーザが同時に使っても応答時間の低下が少ないという長所がある。

分散トランザクション処理(distributed transaction processing)

分散型データベースを更新するトランザクション処理のこと。ひとつのトランザクション処理は複数のデータを更新するのが一般的である。その場合，処理の一貫性が要求され，もし途中で失敗したら元の状態に戻さなければならない。分散トランザクション処理では，複数のデータがネットワークを介した別々のサーバに存在することも考えられる。スループットの向上と処理の一貫性という相反する要求を満たす必要がある。そのため，高度な同時実行制御やコミットメント制御が要求される。
→(関)二相コミットメント



分散ファイルシステム(distributed file system)

複数のシステムにまたがって実現するファイルシステム。システムをネットワークで結び，ネットワークファイルシステム(NFS)やリモートファイルシェアリング(RFS)を用いて，ファイルの物理的な所在を意識することなく自システム内のファイルであるかのようにアクセスできるようにする。すなわちリモートホストのファイルがあたかもローカルホストのファイル同様に扱える。→(関)NFS，RFS，マウント，DCE

分類コード(classification code)

コード化の対象データを，大分類，中分類，小分類などの階層構造に分類し，コードの各桁に対応させ，各分類内で一連番号を付けたコードのこと。データが多くなり，階層構造が深くなると桁数も多くなる。産業分類コード，職業分類コード，勘定科目コードなどがある。グループ分類コード，桁別コードともいう。→(例)グループ分類コード，10進分類コード

分類併合プログラム(sort-merge program)

ユーティリティプログラムのひとつで，ソート(分類)とマージ(併合)の機能をもつプログラム。これらの処理は，通常，自作のプログラムによらず，サービスプログラムとして提供されるソートプログラムやマージプログラムを使用する場合が多い。この場合は入出力ファイルとキー項目の簡単な定義をするだけでよい。→(関)ソート，マージ

平均故障間隔(mean time between failures)

→(参)稼働率

平均故障継続時間(mean time to repair)

→(同)MTTR，(参)可用時間，稼働率

平均サービス時間(mean service time)

1件のトランザクションをサービスするために必要な時間の平均。(1÷平均サービス率)で求める。平均サービス率は，窓口が単位時間にサービスするトランザクションの平均数である。窓口の平均待ち時間を計算するために必要。→(関)サービス時間，窓口数

平均修理時間(mean time to repair)

→(参)稼働率

平均到着間隔(mean time between arrivals)

トランザクションの到着間隔時間の平均。すなわち，あるトランザクションが到着した後，次のトランザクションが到着するまでの時間の平均。これとトランザクション到着間隔の分布が平均待ち時間の決定要因。

平均到着数(average number of arrival)

単位時間にシステムが到着するトランザクションの平均数。実システムでは，時間帯によってこの値が大きく変化する。この数は待ち行列の長さがどのくらいになるかを決定する基準となる。

平均待ち時間(mean time for waiting)

トランザクションがシステムに到着してからサービスを受けるまで待ち行列で待つ時間の平均。ケンドールの記号が示す待ち行列の特性からポラチェック・ヒンチンの公式を使い算出する。

→(関)ポラチェック・ヒンチンの公式

併合(①merge／file merge，②merge／program merge)

ソートされた2本以上の列(データ，レコード，ファイル)を合わせて，1本のソートされた列にすること。マージともいう。

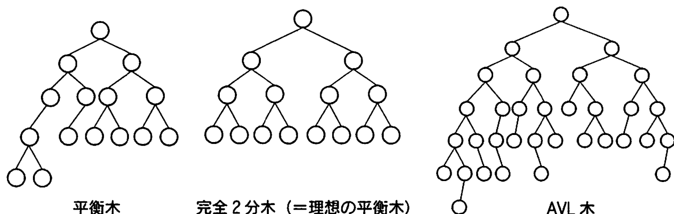
①一定の順序で分類してある(ソートずみの)二つ以上のファイルを，ひとつのファイルに同じルールでまとめること。ファイルのマージという。

②別々に開発した二つ以上のプログラムを全体としてひとつのプログラムにまとめること。プログラムのマージという。

平衡木(balanced tree)

2分探索木の一つで，左右に分かれる枝の先の頂点数がほぼ等しいよう

な形の木構造。はじめはバランスのとれた2分探索木であっても，データの挿入や削除が重なるうちに一方に偏ることが避けられない。探索の効率を考えると，右側と左側のデータ件数がほぼ等しいのが理想的で，極端に偏った場合は平衡木に作り直すことを行う。なお常に左右に2本の枝の出る木を完全2分木という。代表的な平衡木にAVL木がある。



並行処理(concurrent processing／overlap processing)

中央処理装置の遊び時間を有効利用するため，同時に二つ以上の仕事を並行して実行すること。→(同)コンカレント処理

並行テスト(parallel test)

既存システムと新システムを同時に同データで並行的に運用して，新システムの機能，性能を検査すること。処理結果が一致するか，障害が発生しないか，テストで見い出されなかったバグがないかなどの検査に加え，新旧のシステムの性能比較，運用効果の計測などが可能である。

ベータ版(beta version／β version)

完成直前のソフトウェアで，販売前の最終テストの目的で多数のユーザーに供するバージョン。社内テスト用のアルファバージョンに対比する語。多様な環境や局面で使用してもらうことで，未発見のバグや機能不備の情報を集める。製品版の宣伝を兼ねて，パソコン雑誌の付録CD-ROMやパソコン通信で配布するが，ユーザサポートの対象外製品である。→(参考)バージョン

ペーパレス取引(paperless interchange)

企業間ネットワーク，またEDIの普及にともない，企業間の決済が銀行を介して，電子的に行うようになると，従来の伝票や文書で行っていた作業の大部分が不要になる可能性がある。ただし，早期の実現には法的問題，情報化の格差の問題があり，今後の課題といえる。

ヘッダ(header：HDR)

①データやレコードの前に付加した情報(区切りや属性など)。

②外部記憶媒体上の，ファイルの先頭に付けたヘッダラベル。

③コンピュータネットワークで，送信要求があったとき，そのデータを伝送路へ送信するために，プロトコルの各階層でテキストに先行して付加する制御情報。例えば，届け先，発信者の情報，パケット番号，メッセージ番号など。

④プログラムの処理結果や文書をプリント出力するとき，出力内容を見やすくするために各ページの上部，先頭等に印字する標題，見出し，日付，ページ番号などを同一形式，同一内容で印字した行。文書の下部に出力する場合はフッタ(footer)という。

ヘッダラベル(header label)

磁気テープやフロッピーディスク等の媒体上の，ファイルの先頭に付けた文字列。ファイル名等，ファイルの識別，ファイルの管理や保護など，ファイル制御をするための情報をもつ。

ヘルプ機能(help facility)

コンピュータシステムの操作性を高めるために，システムの操作法・エラー対処法などのガイダンスを，ユーザの求めに応じて出力する機能。メニューやアイコンで示した機能の詳細な説明，エラー発生時に対応策のガイダンスを示す機能があると，ユーザはマニュアルを調べる手間が省け，仕事の能率を上げることができる。ヘルプメッセージを吹き出し図形の内部に表示するものを特にバルーン(balloon)ヘルプ，チップ(chip)ヘルプなどという。

変更(①update，②modification，③version up)

①レコードを更新すること。順編成ファイルではファイルを読み込み，更新対象となるレコード内容を書き換え，新たにファイルに出力する。索引順編成ファイルでも，順次アクセスの場合は順編成ファイルと同様，直接アクセスの場合はキーにより，レコードを読み込みレコード内容を書き換える。→(同)更新

②アドレスを修飾すること。

③プログラムのデバッグや機能向上の目的で改作すること。→(同)バージョンアップ，変更作業

変更作業(change／alteration)

年号の変更，金利の変動，税率の変更などのプログラム外部の要因によって，プログラムを変更する作業。運用中のプログラムの変更作業が最優先の作業になる。変更作業は，数多くのプログラムについて実施することが多く，変更が予想される要因については，極力プログラム内に記述しないようにすることも大切である。→(類)改良作業

編集(edit)

①後続の処理のために必要な形式にデータをならべること。例えば，不必要なデータの削除，データの追加，修正，フォーマット変換，レコードのソートなどがある。

②プログラミングでソースプログラムを構成すること。→(参)エディタ

③複数のプログラムを結合してひとつのプログラムとすること。→(参)連係編集プログラム，マージ

④文書処理で，複数の文章を並べ換えたり，画面，紙面に配置すること。

変数(variable)

高水準言語のプログラムで使用できる基本的な単位。記憶場所に対応する識別子で，規格内の任意の値をもちうる可変因子である。固有の名前(変数名)によって識別し，型によって文字，数値，論理値などの種類が決まる。それぞれを文字列変数，数値変数，論理変数という。変数にはプログラム中の文，命令で値を与え，また値を引用することもできる。引用や参照ができる範囲により，大域変数，局所変数に分かれる。また，プログラム中で変数のもつ役割により，制御変数，添字，引数，パラメタ，フラグなどと呼ばれることがある。

→(関)式，型，宣言，(例)局所変数，大域変数，(対)定数，(類)項目

ベンダ(vendor)

メーカおよび販売会社のこと。独自のブランドをもち，ユーザに対し販売した製品に最終的に責任をもつ企業。必要な場合には製品の交換や保障を行う。コンピュータの販売に関わる企業としてはディーラ，ショップもあるが，これらはあらかじめ定めた範囲でユーザに対してサービスを行うことが任務。→(参)サードパーティ

ベンチマークテスト(benchmark test)

異なったコンピュータシステムのハードウェア，ソフトウェアの処理能力を比較，評価するために行う試験。システムの処理能力に対するユーザの要求が異なるため，客観的な尺度は存在しない。通常いくつかの課題が決っており，最もポピュラーな機種を基準にして，要求を意識した各種のテスト用プログラムの処理に要した時間などで能力を比較する。標準的なものとしてバークレイベンチマークやウェットストーンベンチマーク，SPECベンチマーク，TCPベンチマークなどがある。パソコンなどではグラフィック処理能力全般を見るWinTachベンチマークや使用頻度の高いアプリケーションを実行するWinStoneベンチマークなどがある。

ベンチャービジネス(venture business)

既存産業がまだ手がけていない新しい分野に，小資本で創造的に取組み，事業化する冒険的企業。先端技術分野や潜在的なサービス分野には，成功すれば急成長する商品やサービス市場が存在する。しかしこうした分野は実際に挑戦してみなければ結果は分からず，リスクをともなう。安全性を重視する大企業はこうした分野には参入しにくい。先端技術分野であるエレクトロニクス分野はベンチャービジネスの格好な領域で，パソコン関連メーカの中にはベンチャービジネスの出身の企業も多い。

ホ

ポインタ(pointer)

①ディスプレイ画面上で特定の位置を示す矢印などの記号表示。例えばマウスポインタなど。→(関)カーソル

②他のデータを参照するためのデータ。具体的には他のデータの存在場所を示すアドレスである。アドレスポインタともいう。または，これを保持するアドレス部やレジスタ，オブジェクト。

③アクセスするデータやプログラムを格納しているアドレスを保持するレジスタ。→(参)スタック

④データベースなどで使用する技法。レコードとレコードの結び付きを表す印。レコードの特定の項目のうち，次のレコードの記憶位置を示す。ポインタによってレコード間の親子関係を結んでいる。→(参)チェーン

ポイント(point)

①活字の大きさを表す単位。1ポイントは72分の1インチ(0.353mm)で，活字の1辺の長さで文字の大きさを表現する。ポイント数が大きくなれば文字が大きくなる。

②マウスポイント，アクセスポイント，ピリオド(小数点)などの俗称。

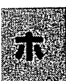
報告書設計(report design)

人間の読める形式に，データをファイルから出力媒体に変換する際の事項を設計する工程。出力項目と媒体を決定する。報告書の利用目的，印字装置の制約などを考え，用紙サイズ，レイアウト，プレプリントの有無，フォントなどをくふうする。その基本となる帳票は次の点に注意して設計する。

(1)必要十分なデータを用紙に納まるようにする
(2)出力項目をユーザにわかりやすい体裁にする
(3)関連帳票との記入位置，記入項目を統一する
(4)ファイリングしやすい形にする

放送衛星(direct broadcasting satellite：DBS)

放送を目的にした人工衛星。BSともいう。通常，赤道上約36000kmの軌道を回る静止衛星である。衛星放送のみを目的としているため，通信衛星(CS)とは区別する。放送衛星は地形や建物の影響を受けにくいため，難視聴地域解消や高品質の放送の決め手として期待されている。しかし，チャネル数の制限，降雨などの天候の影響を受けるといった弱点もある。



ポータビリティ(portability)

→(同)移植性

ホームバンキング(home banking)

家庭内のパソコンや端末で銀行のホストコンピュータにアクセスして受ける銀行サービス。振込，振替，残高照会などのほか，ホームショッピングやチケット予約などのサービス提供がある。

ホームディレクトリ(home directory)

ユーザが利用できる最上位のディレクトリ。シングルユーザのシステムではルートディレクトリであり，マルチユーザのシステムでは，各ユーザに割り当てられたサブディレクトリになる。→(類)ルートディレクトリ

ホームページ(home page)

インターネットのWWWサイトに接続した時に最初に表示される画面。運営する主体が提供するガイダンスページが多く，続けて利用できるページの概要や目次，索引などを記載する。

ホームレコード(home record)

→(参)シノニム

保守(maintenance)

広義には，開発したシステムの最終工程の作業全般をさす。通常は運用中のシステムのトラブルを抑制，回避する作業をさす。保守の対象により次のような保守作業がある。

(1)システムの保守：障害の発生したシステムを回復させること。緊急度が高く，バグの除去やプログラムの修正作業がある。
(2)プログラムの保守：システムが時代の変化に対応できなくなり，ユーザの改良要求にしたがって，機能追加，ファイル構成の変更，容量の

増大などを行う改良作業がある。→(参)バージョン

(3)完全化保守：システムの設定値の変更にともなう変更作業。例えば，年号や基本給の変更，税率や運賃の改訂など外部要因による変更は多い。

(4)適応保守：新しいコンピュータシステムにプログラムを移植すること。

(5)ファイルの保守：ファイル中のレコードの状態を最新に保つために，追加・変更・削除を行うこと。→(参)更新

母集団(population)

調査研究の対象となるもの全体。母集団の平均値を母平均，母集団の分散を母分散という。これらの母集団の性質を示すものを母数(population parameter)という。母数は統計量から推定される。

保守作業(maintenance)

ユーザによって運用中のプログラムの欠陥を修復する作業全般をさす。保守の要求内容によって，(1)修正作業，(2)変更作業，(3)改良作業に分けて考える。プログラムの保守では，残してあるドキュメントに保守の経過を記録し，ユーザへの変更点の説明が重要である。

保守性(serviceability／maintainability)

①コンピュータシステムの総合的な信頼性を評価する際の基準となるRASの要素のひとつ。システムが故障した時の修理のしやすさのこと。平均修理時間(MTTR)などの具体的数値を用いて評価することができ，短い時間であるほど保守性のよいシステムといえる。→(関)RAS，予防保全

②ソフトウェア品質の目やすのひとつで，(1)テストがしやすいこと，(2)プログラムの構造や記述がわかりやすいこと，(3)更新や拡張がしやすいことなどをさす。→(関)ソフトウェア品質

保守体制(field support system)

トラブルの発生に迅速に対応，システムを修理，維持するための組織や方法。保守技術者を専任制にしたり，トラブルの警告機能を設けたり，故障診断プログラムを開発するなど保守技術の改善を図ること。

ホストコンピュータ(host computer)

オンラインシステムなど複数のコンピュータで構成したシステムの中で全体の処理の中核となるコンピュータ。センタコンピュータ，あるいは単にホストともいう。一般的には大型の汎用コンピュータで，大容量のファイル装置や高速プリンタを接続し，ネットワークで結ばれた他のコンピュータに対し種々のサービスを提供するコンピュータ。クライアントサーバモデルでは，単にネットワーク上のコンピュータをすべてホストと呼ぶこ

とがある。いっぽう，通信回線を利用してホストコンピュータに外部からデータを入出力する装置を端末という。

ポストスクリプト(postscript／PostScript)

文書や図形などを紙に印刷したり画面に表示するために加工・編集する方式。デスクトップパブリッシングでよく用い，印刷・表示するページを出力装置に依存することなく記述できる。アメリカのアドビシステムズ社が1985年に開発したPostScriptを各社のレーザプリンタが搭載するようになり，一般的な呼び名になった。

ポストモーテムダンプ(postmortem dump)

プログラムのテストランの最中に，プログラムが異常終了した時点でメモリの内容や変数の値を出力すること。→(関)ダンプ

ホ

保存ファイル(retention file／master file)

捨てたり，廃棄しないで保存しておくべきファイル。情報処理の分野では，例えば，ソースプログラムのリスト，マニュアルの文書ファイルをはじめ，法や内規で保存が定められたデータファイルなど。なお，比較的長期間にわたって継続的に利用されるマスタファイルを保存ファイルと呼ぶこともある。→(類)アーカイブファイル，マスタファイル

ホットスタンバイ方式(hot stand-by system)

障害発生から復旧までの時間を短縮する技法。デュプレックスシステムを使い，主系コンピュータに障害が発生したとき，直ちに待機系コンピュータが対応できるようにすること。→(対)コールドスタンバイ方式

ポップアップ(pop up)

スタックが記憶しているデータを取り出すこと。単にポップ(跳ね上げ)ともいう。→(参)スタック

ボトムアップテスト(bottom up test)

トップダウンテストとは逆に，階層構造化したモジュールの，最下位レベルからテストを始め，順次上位のモジュールと結合してテストする方法。下位モジュールをテストする場合，その上位レベルのモジュールが未完成であるときは，ドライバと呼ぶテスト用プログラムを使用する。ボトムアップテストでは，プログラム開発とテストが初期段階から並行して行うことができ，スケジュール管理もしやすい。しかし，本稼働と同じ条件のテストが最終段階までできないという欠点がある。→(参)ドライバ，(対)トップダウンテスト

ボトムアッププログラミング(bottom up programming)

トップダウンプログラミングとは逆に，最初に小規模なモジュールを考

えて，これらを組み合わせ次第に大きな手順とし，最後に全体を構成する方式。ただし，これはトップダウンプログラミングに対比する概念であって，実際のプログラミングはボトムアップ方式でいくことはほとんどない。システム開発の下流に位置づけられるプログラム言語やハードウェア性能を無視したトップダウンプログラミングを警告する意味合いで，逆に具体的な実現可能性を検討するべきとする概念である。

ボトルネック(bottle neck)

①コンピュータシステム全体の資源の中で最も負荷の高い資源。溢路ともいう。システム全体の処理効率を考えるとき，他の資源に遊び時間が生じるため，この部分を緩和できれば全体の効率があげられる。びんの狭い口の部分が，中身の自由な出入りを妨げることから，流れの妨げとなる部分をさす。

②システム開発等の際に，その開発の妨げとなる事象。→(参)日程計画，クリティカルパス

③プログラム内蔵方式のコンピュータでは，実行の際に，命令とデータが記憶装置とレジスタの間で頻繁にやり取りされる。このことが処理の高速化の妨げになるとして，「フォン・ノイマンのボトルネック」という。

ボリューム(volume)

磁気テープ1巻・磁気ディスク1個・フロッピーディスク1枚のこと。通常，ひとつのボリュームには複数のファイルを記憶することができる。

ボリューム(volume)

磁気テープや磁気ディスクなど，ファイルを格納する媒体を管理する単位。ボリュームにはボリューム名を付け，入出力媒体を特定するために使用する。

ボリューム	単一ファイル	複数ファイル
単一	シングルファイルボリューム	マルチファイルボリューム
複数	マルチボリュームファイル	マルチボリュームマルチファイル

ひとつのボリュームにひとつのファイルを格納したものを単一(シングル)ボリュームファイル(または単一(シングル)ファイルボリューム)，ひとつのボリュームに複数のファイルを格納したものをマルチファイルボリューム，という。その他，表のような呼び方がある。ボリュームにはボリューム名を付け，入出力装置を特定するために使用する。

ボリュームマップ(volume map)

ボリュームはファイルを格納する媒体。マップは，もともとは地図や星

座図を意味し，実態を別の表に対応させた(写像した)図表をさす。ボリュームマップ(記憶媒体の地図)は，記憶媒体の中でプログラムが占有するアドレスの印字出力である。大規模なシステム開発では，たくさんのモジュールを並行して作成し，多数のプログラマが再使用やデバックのためにアクセスするため，統一的な管理が必要である。このためにファイルとボリュームの関係を明確にする必要があり，ボリュームマップを作成する。

ボリュームラベル(volume label)

ボリュームやファイル等を識別するために，データの集合につけた文字列。管理や処理の便宜のために，その集合に関する情報を保持する。磁気テープや磁気ディスクパックは，ボリュームの先頭にボリューム通し番号とVTOCのアドレスなど記録ファイルの管理情報を記録する。各ファイルの先頭にはファイルラベル(ヘッダラベル)をつけて，ファイルの情報を記述する。ファイル末尾にはファイル終わりラベル(EOF)を記録する。記憶媒体によって情報交換を行うため，ISOでは，各媒体のファイル構造を制定しており，この中にラベルの書式も規定している。

ホワイトボックステスト(white box test)

プログラムのテスト法のひとつで，テストデータを選定・作成する際に，プログラムの内部構造・制御の流れに基づいて，すべてのモジュール・ルーチン・文を実行できるように選ぶ方式。つまりプログラムのすべての局面を網羅的にテストするねらいをもっている。構造テスト法ともいう。モジュールテスト(単体テスト)の段階で行うことが多い。しかし，すべてのルーチンを完全に網羅することは困難で，次のような点に重点をおいて，テストデータを作成する。

(1)命令網羅：プログラム中の命令が少なくとも1回は実行される。

(2)判定条件網羅：プログラム中の判定条件が真偽の結果をもち，すべての分岐命令が少なくとも1回は実行される。

(3)複数条件網羅：プログラム中の条件付き分岐命令が判定するすべての可能性を組合せ，すべての命令が少なくとも1回は実行される。

翻訳(translate)

広義には，ある言語を別の言語に変形すること。狭義には以下のように使い分ける。

(1)translate：英語を日本語に訳すなどの国語間の変換，仮想アドレスから実アドレスを算出するなどのアドレス変換。

(2)compile：ソースプログラムを実行可能なオブジェクトプログラムに変換すること。プリプロセッサ，コンパイラなどを用いて翻訳する。

→(参)コンパイル，コンパイラ，プリプロセッサ

(3)interpret：ソースプログラムの1行，または1文(あるいは相当する中間コード)の逐次的なオブジェクトプログラムへの変換。コンパイルと区別して「解釈」ということが多い。→(参)インタプリタ

ホ

マージ(merge)

順序のある2本以上の列を合わせて，1本の整列した列にすること。併合ともいう。

①一定の順序で並んでいる(ソートずみの)複数のファイルを，ひとつのファイルに同じルールでまとめあげること。ファイルのマージという。

②別々に開発した二つ以上のプログラムを全体として一つのプログラムにまとめること。プログラムのマージという。

マージソート(merge sort)

マージの技法を応用して行うソートのアルゴリズム。外部ソートに使われることが多い。基本的な考えは，複数の列の一端から一つずつデータを取り出し，比較し最も小さいデータを別に用意した列に移す。記録したデータの列の一端からデータを取り出し，再び複数のデータを比較し，最も小さいデータを別の列の次に移す。これを繰り返していけば，別の列に全データのソートが完了する。

マイクロメインフレームリンク(micro-mainframe link：MML)

汎用大型コンピュータ(mainframe)とパソコンやワークステーション(microprocessor)をネットワークで接続して利用するシステム形態。こうした使い方をするには汎用大型機と小型機の両方に，マイクロメインフレームリンク用のソフトを備える必要がある。大容量のデータを，効率よく使いやすいユーザインタフェースで，処理することができる。また処理する仕事によっては汎用機とパソコンなど小型機を分離して使うこともでき，ユーザにとっては柔軟性の高いシステムになる。

マウント(mount)

ネットワークファイルシステム(NFS)において，ファイルサーバのファイルシステムをクライアントのあるディレクトリ下に見えるようにする操作。マウント操作を行ったファイルシステムに対してのみNFSの諸機能が使用できる。

マクロ(macro)

①「巨大な」,「巨視的な」等を意味する接頭語。

②「マクロ命令」,「マクロ機能」,「マクロ言語」などの略語。

マクロアセンブラ(macro assembler)

プログラム上でよく使う一連の処理(命令群)を，まとめてひとつの命令として定義し，ほかの命令と同様に使えるような機能を追加したアセンブ

ラ。一連の処理をまとめた命令をマクロ命令といい，このような言語をマクロアセンブラ言語という。アセンブルの際には，マクロ命令は機械語の命令群を組込み，一般の命令同様に機械語コードに変換する。

マクロ機能(macro capability)

アプリケーション操作のコマンドを記憶しておき，ひとつのコマンドで実行する機能。同様の操作コマンドを再実行したり，繰り返したり，一部を変えて実行したいときに，長いコマンドを再入力する手間が省ける。コマンドの記憶には，ソフトによりさまざまな方式がある。

(1)あらかじめ複数のコマンドを登録できる。条件分岐指定も可能である

(2)操作中のコマンド入力を自動的に記憶し，繰り返す機能をもつ

(3)キー入力でなくアイコンを操作していても，コンピュータシステムは，ユーザの操作をコマンドとパラメタに置き換えて実行し，記憶する

(4)プログラム言語の機能として組み込まれており，プログラムで指定できるものもある

マクロ言語(macro language)



マクロ機能の強化，アプリケーション記述との統一化を図ったマクロ命令の記述用言語。アプリケーションプログラムの自動運転のプログラムだけでなく，条件分岐などを含んだ複雑なプログラミングも可能である。マイクロソフト社のVisual Basic Programming System Applications Edition，ロータス社のLotusScript，アップル社のAppleScriptなどは，自社のアプリケーション間で共通に使用できるマクロ機能を提供している。

マクロ命令(macro instruction)

①ソースプログラムを簡単に書くために，複数の命令で構成する処理を1命令(パラメタ指定を含んでもよい)で置き換えた命令。

②ソースプログラムの中で，特別な意味をもつ命令で一連の機械語命令と等価な命令。アセンブラのマクロ命令がこれに相当する。

③原始言語で記述された命令の中で，あらかじめ同じ原始言語で定義した一連の命令に置き換える命令。一連の処理を定義することをマクロ定義といい，マクロ定義したルーチンを翻訳前に挿入することをマクロ展開という。→(参)プリプロセッサ，プリコンパイラ

マスタ索引(master index)

シリンダ索引の索引。シリンダ索引が大きくなり，検索時間を要する場合に作成され，シリンダ索引のトラックの最大キーとトラック番号が記録される。

マスタファイル(master file)

比較的長期間にわたって保存し，しばしば参照するファイルで，更新しながら継続的に使用するファイル。台帳として使われるファイルに相当する。基本ファイル，永久ファイル，保存ファイルとも呼ぶ。これに対しデータ発生のつど作るファイルをトランザクションファイル，発生ファイル，一時ファイルなどという。

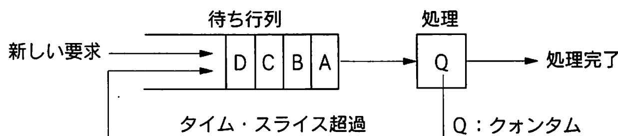
待ち行列(queue)

何かを待つための行列，あるいはそのような行列を作ること。

(1)OR(operations research)の分野では，文字どおりサービスを待つ人や荷，自動車の列などに例えられる。

(2)情報処理の分野では，先入れ先出し(FIFO)のデータ構造。複数のデータを順次に記憶するが，取り出せるデータは常に1件で，最も早く記憶したデータのみである。列の一端で追加し，他の端で取り出しが可能である。コンピュータによる処理を待つジョブの列やマルチプログラミングで処理する時を待つジョブ，出力装置が空くのを待つジョブの列(出力待ち行列)などがある。→(参)FIFO

(3)データ通信の分野では，蓄積交換回線で回線の空きや相手先のバッファの空きを待つパケットやメッセージが待ち行列を作る。



待ち行列理論(queuing theory)

あるサービスを受けるために待つ現象，およびサービスを受けるのに要する時間を確率的にとらえる理論。例えば，銀行や病院の受付を待つ人の列がどの程度の長さになり，したがってどの程度待合室に椅子を用意すべきか，また，サービスを行う受付窓口数をどの程度増やせば経済的か，などが問題になる。待ち行列理論は，この問題を数学的に分析しようとしたもの。計算の基本となるものは，次の3つである。

(1)サービスを受けようとするものの到着分布

(2)1回のサービスを受けるのに必要となる時間分布

(3)サービスの窓口数

情報・通信の分野では，トランザクション処理の計算機待ち時間，データ通信における回線待ち時間などを計算し，システム設計に利用する。

待ち時間(latency／waiting time／delay)

ある動作の要求があった瞬間からその動作を開始する瞬間までの時間。

①磁気ディスク装置に対する読み書きの要求をした時，アクセスアームを所定の場所に位置づけ(位置決め時間)，特定のデータの記録場所が回転してきた後(回転待ち時間)読み書きが始まる。読み書きが始まるまでの時間を待ち時間という。

②スプーリングを行うシステムで，ジョブを投入してから実行するまでの時間。

③オンラインシステムで，トランザクションの要求をしてから応答が返ってくるまでの時間。

マップ(map)

広義には，ある実態を別の表に対応させた(写像した)図表。もともとは地図や星座図の意味。狭義には，メモリマップ(記憶域地図，記憶領域図)をさす。主記憶の中でプログラムが占有する領域のアドレスを表したものである。デバッグ時に変数や配列の領域を調べて，ダンプする範囲を決める場合などに使う。



窓口数(number of server)

トランザクションを受けつける窓口の数。窓口を設けるコストと待ち時間を考慮して決定する。待ち時間についての基準(トランザクションのうちm個は待ち時間がn分以内であることなど)により決めることが多い。ケンドールの記号では，最後に指定する。　M／M／□　←窓口数

マトリックス(matrix)

①mn個の要素を，水平方向のm個の行(row)と水平方向のn個の列(column)に並べた配列。プログラム言語で取り扱う場合は，2次元配列(2次元テーブル)といい，行と列の位置を添字で指定したとき，要素が一意的に定まれば，要素自体は物理的に長方形に並んでいなくてもよい。

②同一の部品要素を行と列をなすように格子状に配置したもの。例えば，表示や印字で使うドットの集まりをドットマトリックスという。磁気コアを格子状に配置した記憶媒体はコアマトリックスと呼ぶ。

マニュアル(manual)

①コンピュータ資源を使うオペレータ，プログラマ，ユーザのための取扱い説明書。手順，命令，操作法などのほか，エラー処理，トラブル処理の対処法なども含む。原義は便覧，案内の意。→(類)利用者マニュアル

②自動(オートマチック)処理に対比して，手動・手作業(マニュアル)によ

る処理のこと。

マルチファイルボリューム(multifile volume)

磁気テープで，ひとつのボリューム上に複数個のファイルをもつ媒体。逆にファイルが大きいため複数のボリュームにまたがって記憶しているファイルをマルチボリュームファイルという。→(関)ボリューム

マルチプラットフォーム(multi platform)

複数の動作環境上で実行可能なシステムの性質のこと。マルチプラットフォームシステムでは，各動作環境とインタフェースを合わせた中間のソフトウェア(ミドルウェア)をそれぞれ用意する。そのソフトウェアを動作環境に合わせて差し替えることによって，複数の動作環境で動作可能としている。アプリケーションをマルチプラットフォーム化するためのミドルウェアとしては，データベース管理システム，通信管理システム，GUIプログラムなどがある。



マルチベンダ(multi-vendor)

複数の異なるメーカやベンダ(販売元)の製品がひとつのシステム内で同時に用いられること。必ずしも明確な定義があるわけではない。初期の情報システムでは，周辺機器も本体メーカの供給する製品でそろえるシングルベンダの形態が多かった。しかし，分散処理システム，オープンシステムが普及するに従い，端末や通信制御装置など汎用コンピュータ以外の製品に，別のベンダからコストパフォーマンスのよいものがあらわれた。ユーザは徐々に複数メーカの製品を受入れ，マルチベンダ化が進んできた。

マルチボリュームファイル(multivolume file)

非常に大きなファイルで，ひとつのボリューム上に記録できず，複数個のボリュームにまたがって記録されるファイル。→(関)ボリューム

マルチメディア処理(multimedia processing)

マルチメディアデータの制作，通信，出力などの処理。コンピュータ技術の進歩により，文字情報，映像情報，音声情報などがディジタル情報として扱うことが可能になり，対話機構を備えることで従来のメディアにない機能をもった統合的な情報伝達メディアになる。

マルチメディアデータベース(multimedia database)

従来のデータベースが扱うデータは主として数字や文字で表現してあった。これに対して，近年進歩した図形，静止画像，動画，音声などが混在する多様なデータを蓄積，操作できるようにしたデータベースをさす。

マンマシンインタフェース(man machine interface : MMI)

→(同)ヒューマンインタフェース

見出しラベル(header label：HDR label／beginning of file label：BOF)

磁気テープ装置にデータファイルを収容する際，ファイルの先頭に記録し，ファイルを識別するための特別なレコード。通常の台帳の表紙の部分に相当し，ファイル名，ファイル番号，作成年月日，有効期限などの情報を記憶する。

ミラー(mirror)

定期的なコピー。または，同じデータを同時に2か所のディスクに書くこと。一方が読めなくなったとき，他方が利用できる。

ミラーサーバ(mirror server)

同等のサービスを行うもうひとつ以上のサーバ。ミラーサーバと呼ぶ場合，サーバの処理機能が同等であるだけでなく，内部で持つデータも同等であることが多い。クライアントサーバシステムで，クライアントからある特定のサーバに対する要求が集中するとコンピュータや通信ネットワークの負荷が偏り，システム全体のスループットが大きく下がる。そこで，多くの要求が集中すると予想されるサーバは，いくつかのミラーサーバを置き，サーバや通信ネットワークの負荷の分散を図る。

ミラーサイト(mirror site)

インターネットの同一WWW，FTP，Gopherなどのサイトを複数のサーバに置いたもの。ミラーサイトを設置する目的は，最も近い場所にアクセスすることによるメインサイトサーバの負荷分散，応答時間の短縮，通信費の削減などである。非常にアクセス回数の多いサーチエンジンなどでは，ミラーサイトを用意して検索スピードの向上を目指している。
→(参)ミラーリング，レプリケーション

ミラーディスク(mirror disk)

障害に備えて，主系とは別に予備用としてデータを記憶するディスク。バックアップと異なり，処理のつど，主系と同様の処理を行う。この意味でミラーディスクログということもある。→(類)バックアップファイル

ミラーリング(mirroring)

①鏡に写すように同一情報を別の場所に持つこと。ディスクのミラーリングとは同一情報を別ディスク領域に常に保持した状態にすることで，ディスクの二重化とほぼ同じ意味である。ミラーリングによって，一方のデータがディスククラッシュなどで破壊されても，他方に残っているデータを使用できるため，信頼性が非常に高くなる。→(関)ミラーディスク

②インターネットでは，同じ情報を別のサーバにコピーしておき，どちらにアクセスしても同じ情報を得られるようにすること。ミラーリングしている場所が遠隔にある場合，利用者がデータをアクセスする際に近い方をアクセスすればよく，通信コストが節約できる。この場合のミラーリングはフォールトトレラントの目的よりも，アクセス速度を向上させるキャッシュに近い目的である。

ム・メ

無条件飛越し(unconditional jump)

プログラムのそれまでの条件コードや状態語とは無関係に，他の部分へ制御を移すこと。無条件分岐命令ともいう。

メインテナンスフリー(maintenance free)

保守不要のこと，あるいは保守契約のないシステム。コンピュータシステムの維持には保守は必要である。しかしコンピュータの小型化・低価格化が進み，オフィスや一般家庭まで普及するようになった状況では，大量のマシンをメーカが個別に保守することはコストがかかる。そこで価格と保守費用を下げるため，製品自体を故障がおきにくい設計，仕様にして，保守作業を最大限省けるようにする。

メインフレーマ(mainframer)

大型や中型の汎用コンピュータを開発・製造しているメーカ。これらのコンピュータのことをメインフレームという。日本では富士通，日本電気，日立製作所など，アメリカではIBM，UNISYSなど。かつては大型汎用機が中心的な存在だったが，小型で高性能なCPUチップの開発などプロセッサ技術の進歩にともない，コンピュータの小型化が進み，現在はパソコンやワークステーションがコンピュータ市場の主役になりつつある。

メインフレーム(mainframe)

用途を特定しないコンピュータ(汎用コンピュータ)のひとつ。大型から小型のものまで規模はいろいろあるが，通常は運用保守の常駐要員を必要とする規模のクラスをさす。一般的には，ある機関や会社のホストマシンとして事務処理や科学技術計算の中核に使用する大型コンピュータのこと

をさす。近年は，既存のソフトやデータ資産を蓄積，活用するデータベースマシンとして位置づけられているため，ハイエンドサーバと呼ぶこともある。→(類)汎用コンピュータ

メインプログラム(main program)

プログラムの中心になる部分。主プログラムともいう。プログラムは，中心となる部分(メインルーチン，制御セグメント，上位モジュール)とそこから呼び出す部分(サブルーチン，従属セグメント，下位モジュール)からなる。小さなプログラムではメインプログラムだけで作成することもあるが，多くのプログラムはメインプログラムとサブプログラムから作る。プログラムの実行にはメインプログラムが必ずなければならない。

メインルーチン(main routine)

メインプログラムと同じ意味。プログラムの骨格になる部分。主ルーチンともいう。　オブジェクト指向プログラミングにおいて，オブジェクトの実行する手続き。オブジェクトにメッセージを送ると，メッセージに対応するメソッドがさがしだされ，実行される。

メタデータ(metadata)

データそのものではなく，データの意味を記述するデータのこと。データの操作を定義する前提となる。例えば，(6，6，24)はデータそのものであるが，これが(学年，組，学生番号)であるのか(年，月，日)であるのかの意味がメタデータである。通常，データの意味は，それを扱うプログラムやシステムが定義する。DBMSでは，データとメタデータの双方をデータとして扱うので，メタデータのメタデータを定義しておく。これを記録しておくのがシステムカタログ(system catalog)またはDD／Dである。

メッセージ(message)；

広義には，2つのシステム間，プロセス間で交わすデータ。OSとアプリケーション間で交わすデータも含む。

①人間がコンピュータシステムに与えたコマンド(指示)とコンピュータから人間への応答，通信文。ワーニングメッセージ，処理の終了メッセージなど。対話型処理では，広義の応答，つまり選択可能なメニューやアイコン，入力可能状態を示すプロンプトマークやエラーメッセージ，ワーニングメッセージなどもメッセージである。

②オペレーティングシステムからアプリケーションプログラムに与える処理要求。例えばWindowsでは，アプリケーションプログラム制御の主導権はWindows本体が持ち，アプリケーションプログラムはWindowsから順不同で渡されるメッセージに対応できるよう作られる。

③オブジェクト指向プログラミングにおいては，オブジェクト間で交わされるデータ。
④データ通信においては，2地点以上の局間で通信するデータ。
⑤計算機内の処理機能単位(プロセス)間で通信しあう情報。

メディア変換(medium conversion)

①広義には，データの内容は変えずに異なる形式に変換すること。例えば，テキストファイルの内容をファクシミリ通信の信号や音声出力装置への信号に変換すること。
②情報処理では，データの内容を変えずに異なる媒体に変換すること。単純な複写ではなく，例えば，磁気テープから磁気ディスクへ，フロッピーディスクからMOディスクへと，記録内容を他の媒体物を移すこと。
③マルチメディアの分野では，従来アナログ情報として記録されていたデータをディジタル情報として記録し直すこと。例えば，アナログ録音のテープからCD用のディジタルデータへ変換することなどを含む。

メニュー(menu)

料理の献立表(メニュー)と同じように，ユーザが選択指定可能なプログラムやファイルを一覧表にして表示した画面。パソコンなどでデータを処理する場合，作業の手順や内容を画面に表示して利用者に選択させる。

階層型メニューは，選択項目が多い場合に選択肢に階層構造を持たせたもので，メインメニュー(mainmenu)，サブメニュー(submenu)などに階層化してある。また，メニューを常時，画面の上部(または下部)に表示する行をメニューバー(menu bar)という。メニュー選択には，次のような方法がある。

(1)画面に示された選択コード(メニューの行番号など)を入力する
(2)選択機能の一文字(頭文字など)を入力する
(3)選択するメニューをマウスでクリックする
(4)指定のプログラムファンクションキーを押す

メモリダンプ(memory dump)

デバッギングの際に，必要により，主記憶メモリの内容を出力すること。

面接(intervie／interviewing)

広義には，情報収集，情報交換，評価等のために，調査対象者と直接対面し，対話，観察等によって情報の提供や交換を行うこと。インタビューともいう。システムエンジニア，アドミニストレータ業務に関する面接は，主として現状分析やシステム構築のために行う取材で，業務の内容や

事情聴取，仕事上の不満や苦情，提案などを聞くために行う。部門内の業務の改変に関することなので，質問・応答の際には相互の理解と意思疎通が必要である。

メンバ(member)

①区分編成ファイルを構成する互いに独立したデータ群。区分編成ファイルは直接アクセス記憶装置上に存在し，いくつかの互いに独立したデータ群(子レコード)の部分と，おのおののデータ群に関する情報を持つ登録簿で構成する。メンバ内の編成は順編成ファイルと同じである。

②データベースでは，異なるレコード間のつながりの構造を表すためにひとつの親レコード(owner)と複数の子レコード(member)をセットと呼んで定義する。データベース設計者は，この構造を利用してデータの蓄積，管理，呼出し，結合を行う。

モ

モ

モード(mode)

①統計用語で最頻値のこと。等しい幅の階級に分けた度数分布表で度数が最も多いところの階級値。

②あることがらにいくつかの排他的な方式や種類があるとき，それぞれの方式や種類をモードという。例えば，次のような例がある。

(1)キーボードからの入力方式で，カナ入力，ローマ字入力などの指定

(2)ディスプレイ画面のグラフィックモードで，出力する色数などの指定

(3)フロッピーディスクのフォーマットで，容量1.44メガバイト，1.2メガバイト，720キロバイトなどの指定

(4)データ通信の際のコンテンション方式，ポーリング・セレクティング方式などの指定

目的プログラム(target program)

→(同)オブジェクトプログラム

文字演算子(character operator)

プログラム中で文字型のデータを対象に演算を施すときの記号。

文字多重放送(teletext)

テレビ電波のすきま(垂直帰線消去期間)を利用して，文字や図形情報を送信する放送。聴力障害者のための字幕サービスのように画面を補完する

利用，ニュース，天気予報，交通情報など速報的な情報を画面に出す利用ができる。→(参)音声多重放送

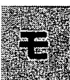
文字定数(non-numeric literal)

文字列で構成する定数。両端を引用符で囲んで表現する。計算の対象にできない。プログラム言語により，文字列長の制約，引用符自身の含め方は異なる。

モジュール(module)

①原義は，(1)測定の単位・標準，(2)全体の中に組み込む部分または組込みの基準尺度。

①ハードウェアではメモリや各種の回路を機能単位に分割し，組込みや増設が容易に行えるようにまとめた単位をモジュールという。モジュールは単体で定めた機能を有し，より大きな単位の部品，付属品の形態になる。ひとつのモジュールは単位として取り除き，別のモジュールに取り替えることができる。モジュールで構成したハードウェアは，拡張性，融通性，保守性にすぐれた特性をもつ。



②構造化プログラミングでは，ある機能を有する小プログラムをモジュールという。容易に交換し得るような機能単位のプログラムをさす。大きなシステムを構築する場合には，プログラムをモジュール化して作成すれば，次のような利点があり開発効率の向上が期待できる。

(1)開発時に分業化できる

(2)誤りの発生部分を特定できる

(3)機能改善に部分変更で対応しやすい。

さらに，小規模なソースプログラムの中でも，ある機能を実行する小部分ごとに分けて，プログラミングの能率を上げ，可読性を高める。構造化プログラミングの普及により，プログラム開発を各機能ごとに部分に分けて行うようになった。

→(参)トップダウンプログラミング，構造化プログラミング

③ソフトウェア開発の段階では，プログラム単位を表す用語で，コンパイル前のソースプログラム(原始プログラム)をソースモジュール，コンパイラの出力プログラム(目的プログラム)をオブジェクトモジュール，リンカの出力プログラム(実行形式プログラム)をロードモジュールという。

④ひとまとまりの手続きやデータの宣言からなる言語の構成要素で，他の同様の構成要素と相互に作用するプログラム単位。副プログラムやサブルーチンなど個別にコンパイルできる範囲をさす。例えば，呼び出すプ

ログラム，呼び出されるプログラムはそれぞれモジュールである。

モジュール化(modularization)

大規模なプログラムを開発する際に，プログラムを各機能ごとにいくつかの小部分に分けて行うこと。分割の方式やサイズは，システムで異なり一定していないが，処理機能やデータ構造から判断して，独立性の高い単位に分割する。主要な分割技法には次のようなものがある。

(1)源泉／変換／吸収(STS)分割法
(2)トランザクション分割法
(3)共通機能分割法
(4)データ構造分割法

モジュール間インタフェース(interface between modules)

一般的には，プログラム間インタフェースと同義。構造化設計の段階で分割するモジュールの相互間で授受するデータの定義といえる。すなわち，下位(従属)モジュールが上位モジュールからどんな情報を受け取り，どんな情報を返すかを明確にすることで，モジュール間インタフェース定義図で表現する。

モジュール言語(module language)

アプリケーションプログラムが，データベースのデータ操作を要求するためのモジュールを定義する言語処理系。適用業務プログラムからのデータの独立性を保証するために，DBMSとのインタフェースを果たすモジュールを設ける。モジュールは手続きのひとつであり，DMLで記述できる。親言語のプログラムが呼出し文で指定すると，モジュール内で定義されたデータベース操作が実行される。

モジュール構造(module structure)

システムをモジュールに分割して構成するときの，論理的な構造。モジュール分割を階層構造的に行えば，プログラム開発は全体から部分へと段階的にトップダウン展開することで実現できる。また，モジュールの独立性を高くすることも重要であり，これはモジュール間結合度を弱く，モジュール独立性を高くする方向で検討する。

モジュール設計(module design)

プログラム設計の段階で行うモジュールの設計。モジュール化，セグメント化にほぼ同義。

モジュールテスト(module test)

プログラムを複数のモジュールで構成するとき，各モジュールの機能単位ごとに独立したテストを行うこと。単体テストともいう。

個々のモジュールは小規模で，機能が限られていることが多く，テストケースの設定・テストデータの作成も比較的容易であり，エラーを検出した場合も，原因の特定・修正は比較的容易である。ただし，各モジュールが単独で実行可能でない場合は，スタブ・ドライバなどのテスト用プログラムが必要になる。単体テスト後はモジュールの結合テストを行い，プログラム間インタフェースや全体としての機能の一致度を検証する。

モジュール分割(module division)

→(同)モジュール化

文字列(character string)

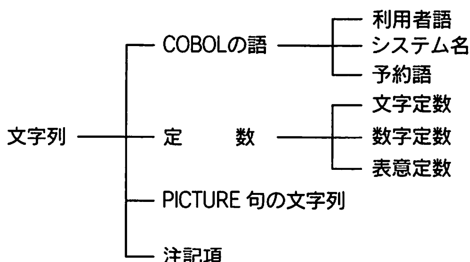
文字だけから成り立っている列。記号，英字(アルファベット)，2進数字などの列のこと。文字を連ねることで，文字数より多くの情報を表す。また人間にわかりやすい一意名を定義できる。文字列処理とは，文字列の集合から，単語の抽出，特定の語の検索などをさす。

文字列(string)

記憶域上の連続する文字列のことで，最後はヌルである。文字列のポインタは先頭の文字へのポインタのことで，インクリメントすると次の文字のアドレスをさす。

文字列(character-string)

ひとつながりの文字の列。次のように分類される。



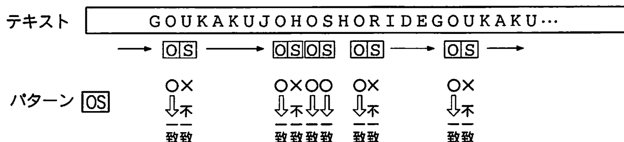
文字列検査(inspect)

データ項目中の文字や文字列の出現回数を数えたり，それらを他の文字や文字列に置き換えたりすることをいう。

文字列の探索(string searching／string pattern matching)

テキスト(文字列長さm)中に，指定の文字列(文字列長さn)と一致する部分があるかを調べること。探索に成功した場合は，その位置を決定する。ワープロやエディタのテキスト編集，データベースからのデータ検索に応用される技法である。簡単な方法には，全テキストを1字ごとにスキ

ャン(走査)する方法があるが，計算量は多くなる(m n に比例)。クヌース-モリス-プラット(Knuth, Morris, Pratt：KMP)のアルゴリズム，ボイヤ-ムーア(Boyer, Moore：BM)のアルゴリズムは，不一致を検出した文字列長さを利用して無駄な比較を避けるアルゴリズムで，計算量は n に比例する。



モデル化(modeling)

広義には，事実，現象，概念についての仮説の体系。部門経営と業績に関するモデル化とは次のようなものがある。

(1)物や情報の流れを，人や組織の相互関係の中で記述するプロセスモデル(記述モデル)

(2)業績に影響する諸変数の変動から業績を予測する予測モデル(数学モデル)

(3)問題について最適解を与える決定モデル(シミュレーションモデル)

戻り値(returned value／return value)

関数や副プログラムに値を引き渡して実行した後，関数や副プログラムから戻される値。返り値ともいう。

戻り番地(return address)

外部関数や副プログラムに制御を移し，その実行が終了した後，制御が戻るべきアドレス。復帰アドレス，返り番地ともいう。

問題解決①(solution of the problem／solution)

①消極的には，障害や問題の原因を除去すること，対応策。積極的には，原因分析の結果をふまえ，改善目標に沿って仕事のシステムを変更する多くの代替案を，評価基準や制約条件に照らして評価し，具体化する作業。解決案の実施には，改善意義，効果等をよく把握し，改善時とその後に生じる新たな問題についても対処法などを考えておく必要がある。

②与えられた問題を解くこと。情報処理では問題をモデルで記述し(形式化して)，有効なアルゴリズムを考えて問題に適用し，解を得ること。通常は問題に有効なアルゴリズムを発見することをさす。

問題発見(recognition of the problem)

問題とは，業績の目標水準(期待水準)と達成水準との間のギャップをさす。常に流動的であるが，急激な変化が現われない場合もある。問題を知

覚しないと組織の進展や革新はならないので，常に諸活動の合理化，生産性の向上を意識して業務の仕組みや技術を見直し，改善する必要がある。こうした視点や方法論を問題発見という。

問題向き言語(problem oriented language：POL)

ある種の問題について解法を記述するために特に適したプログラム言語。ALGOL，FORTRANなどの手続き型言語，GPSS，SIMSCRIPTなどのシミュレーション用言語，LISP，IPLVなどのリスト処理用言語，情報検索用言語等の総称。

モンテカルロ法(Monte Carlo method)

数学的な問題の処理に乱数を用いる方法。シミュレーションの重要な技法のひとつ。乱数列を使うことで，数値計算問題の近似値解を得るもの。確率に関することからカジノで有名なモンテカルロの名前をつけた。

ヤ・ユ・ヨ

有限要素法(finite element method：FEM)

微分方程式の数値解法のひとつ。差分法，残差法ともいう。構造力学とともに発展し，今日では工学の広い分野で利用される。

ユーザID(user identifier／user identification)

利用者個人を識別するために与えたコード(IDコード)，またはIDコードにより利用者個人を識別すること。入退室，ファイルアクセス等の際に資格を明らかにし，利用権を登録するためのコード。第三者が知ることができる利用者名または番号と，第三者が知ることができないパスワードまたは暗証番号からなる。利用者名は現在誰が使用中であるかを調べるためにあり，パスワードは本人確認のためにある。

ユーザインタフェース(user interface)

→(同)ヒューマンインタフェース

ユーザインタフェースサーバ(user interface server)

クライアントサーバシステムの中で，ユーザインタフェース機能を提供するサーバ。ユーザインタフェースにGUIやウインドウ技術を使用すると，端末には負荷の大きいグラフィックス処理が発生する。すると端末の業務に支障が出るので，グラフィックス処理に適した専用サーバが必要である。ウインドウサーバと呼ぶこともある。

ユーザサポート(user support)

ハードウェアやソフトウェアのユーザに対して，操作教育やトラブル対応，情報提供などのサービスを提供し，効果的な運用を支援すること。必ずしもすべてのユーザに均等に無償で行うわけではなく，内容に応じて個別に契約を結ぶのが基本である。

ユーザ登録(registration of user name)

ハードウェアやソフトウェアを購入した際に，同封してあるハガキなどを通じて，製品のシリアル番号および使用機種名，自分の住所，氏名などをメーカに連絡すること。この手続きをもって正式ユーザとして登録される。ユーザ登録は義務ではないが，運用上のトラブル対応などのサポートを受けるためにも必要である。またバージョンアップが行われた際に，登録ユーザに対して新製品を市価よりも割安に提供しているメーカもある。

ユーザファイル(user file)

システムファイルに対し，主としてユーザが作成した応用ソフトウェアやそのデータを記録するファイル。データファイルともいう。その内容はさまざまで，用途・目的・形式によってさらに分類できる。

ユーザフレンドリ(user friendly)

ヒューマンインタフェースを設計する視点のひとつ。エンドユーザの要求に応えるような使いやすさ，分かりやすさのこと。画面構成ではアイコン，メニューなどによる機能表示，マウスや1文字入力による機能選択，入力操作についての分かりやすいメッセージやヘルプ機能などをさす。

ユーザ要件(user's requirements)

システムのユーザ，クライアントが，これから開発するシステムに要求する機能，性能，安全性，障害対策などを把握，整理したもの。開発システムの機能，性能を設定するために必要である。→(関)要求分析

優先順位(precedence of operator)

①複数の演算子間における評価の優先順位。式の中で，複数の演算子を使用する場合には，この優先順位に従って演算が行われる。この優先順位は(　)によって変えることができる。

②スループットの向上や，資源の有効利用を図るため，マルチプログラミング方式を採用して，同時に多数のジョブを処理する。これらの各処理にその必要の度合いに応じ資源の割当てを行う。その割当ての基準を優先順位という。優先順位には，ジョブ待ち優先順位や実行優先順位などがある。優先度ともいう。

ユ

要求定義(requirements definition)

システムのユーザ，クライアントが，これから開発するシステムにどのような機能，性能，安全性，障害対策を要求しているかを把握，整理し，開発システムの機能，性能を設定すること。次のような点に留意して行う。

(1)ユーザ部門の率直な要求を知り得る協力関係を築く
(2)現状への不満を解消する具体的なニーズを把握する
(3)多くの要求を整理し，重点課題を明らかにする
(4)設定した開発システムが要求をどう満たすかフィードバックする
(5)新システム移行，運用にともなう業務の変更点に理解を求める

要求分析(request analysis)

現状分析により新システムの開発が必要と判断した場合，利用部門の要求案と期間・費用・人員などを配慮した開発計画案の双方をまとめること。要求仕様書や開発計画書を作り，利用部門との調整を図る。

容量(capacity)

記憶装置・通信回線などの容量，能力。主記憶装置，補助記憶装置の場合は，記憶可能な最大データ量で，通信媒体の場合は，単位時間に伝送可能な最大データ量で表す。

呼出し命令(call instruction)

サブルーチンに実行の制御を渡す命令。コール命令ともいう。一般にサブルーチンからの戻り番地をスタックに格納して，サブルーチンの入口点に制御を移す。

予防保全(preventive maintenance)

あらかじめ定めた期間内に定期的に点検や動作の確認を行い，障害を未然に防止するための保守作業。予防保全は障害の発生を抑えることとなり，MTBFを向上させ，システムの信頼性に寄与する。例えば，磁気ディスク装置などでときどき発生する読取り障害など，自動的にリカバリできたものについても，OSのログ情報を解析した上で交換することが多い。

読取り専用属性(read only attribute)

ファイル属性のひとつで，ユーザに読出しと参照だけを許す属性。

予約語(reserved word)

各プログラム言語の中で，特定の意味や用法をあらかじめ決めてあり，ユーザが独自の変数名などに使用することができない語(綴り)。

予約サービスシステム(reservation service system)

列車，航空機などの座席予約，ホテル，催しなどのチケット予約，病院の検診や会議室の予約などを行う問合せ応答システム。オンラインリアル

タイム処理の原型ともいえる。代表的な例には，アメリカン航空の座席予約管理システムのセーバ，ユナイテッド航空のアポロがある。日本では，1960年から始まったJR(当時は国鉄)の「みどりの窓口」サービスを行うMARS(マルス)がある。

ラ

ライセンス(license)

使用，占有，アクセスなどに対する許可，許諾。

ライフサイクル(life cycle)

ハードウェアやソフトウェアの導入から廃棄までの期間，またはその間の増設や変更の過程。業務処理などに，いったんコンピュータシステムを利用し始めると，それを使わなかった以前の状態に戻すことは不可能になる。より多くの仕事をコンピュータで処理する要求が生じ，より高性能なハードウェアの増設，より使いやすいソフトウェアへの改善がなされる。

また，コンピュータ技術の進歩は速いため，次々と新しいハードウェアやソフトウェアが開発されてくる。このためユーザは3～5年程度でハードやソフトを刷新することが多い。廃棄されるシステムがあれば代替の新規システムが要求されるわけで，こうしたコンピュータシステムの誕生・成長・廃棄の流れをライフサイクルという。

ライブラリ(library)

本を集め，保管，利用に供する図書館(ライブラリ)にならい，プログラムを体系的に組織し，保管する場所。集めたプログラムそのものをライブラリともいう。ライブラリのプログラムは，プログラマが必要に応じて取り出し，編集し，利用することができる。またシステムが利用するサブルーチンなどは，連係編集やリンクが容易に行えるように相対形式またはソースモジュールになっている。

ラインエディタ(line editor)

行単位にプログラムの編集が可能なプログラム。

ラベル(label)

①プログラム中で飛越し先などを示すためにつける標識名。モジュールや配列領域の先頭に名前をつけるときなどに用いる。

②ボリュームやファイル等のデータの集合に識別用につけた文字列。ラベ

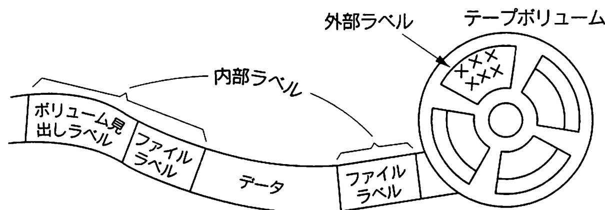
ルはファイルの管理や処理の便宜のため使用し，その集合に関する情報を保持し，識別する。

(1)外部ラベル：ファイル媒体(ボリューム)の外部に貼る名称記入用紙。

(2)内部ラベル：ボリュームに付けるものとファイルに付けるものがある。

・ボリュームラベル：磁気テープや磁気ディスクなどのボリュームの始めと終わりに付けるラベルで，ボリューム内のファイルを識別するための情報などを記述する。

・ファイルラベル：ファイルの始めと終わりに付けるラベルで，ファイルの識別情報・作成日情報などが書かれている。ラベルもまたファイルを構成するレコードであるので，ラベルレコードとも呼ばれる。



欄(①field，②column／field)

①レコード中でデータを記録する物理的な場所。

②表計算ソフトなどで，データを入力する項目欄(セル)の列。→(同)列

ランダマイジング(randomizing)

直接編成ファイルのレコードの位置決めにおいて，レコードの予測される範囲のキー値から算出したアドレスが，磁気ディスクの指定した記録範囲(エクステント)の全域にほぼ一様に分布するようにすること。ハッシングと同義。

ランダムアクセス(random access)

ファイルアクセスの方法のひとつで，ファイルオープン後，ファイル上の位置を指定することによって任意の位置にアクセスすることができる。ファイルにランダムにアクセスできるかどうかは，ファイルの性質に依存する。例えば，ディスク上のファイルはランダムアクセス可能であるが，標準入出力や，テープ上のファイルは不可能である。

ランダム到着(random arrival)

あるトランザクションの到着と別のトランザクションの到着にまったく関連性がない状態。平均到着率が分かっているとき単位時間にランダム到着するトランザクション数の発生確率はポアソン分布に従う。

ランダムプロセッシング(random processing)

ファイルを更新する処理で，必要な個所を直接アクセスして行うこと。磁気ディスクなどのランダムアクセスの可能な記憶装置(DASD)を使うことで可能になる。オンライン処理には不可欠な処理。順次処理に対比して用いる語。乱処理ともいう。

ランニングコスト(running cost)

システムの運用，設備や機械等の運転，稼働にともなって発生する費用。コンピュータシステムの場合，電気代や保守点検費，出力用紙，補助記憶媒体，インクリボンなどの消耗品代など。プログラムのメインテナンスにかかる費用を含むこともある。

乱呼出し(direct access)

直接アクセスにほぼ同義。記憶装置からデータを得たり，記憶装置にデータを入れたりする機能で，その過程が前回アクセスしたデータとは無関係に目的のデータの記憶場所にだけ依存するもの。ランダムアクセスともいう。→(類)直接アクセス，(関)直接編成ファイル，索引順次編成

リ

リアルタイムオペレーティングシステム(real time operating system)

主に制御用コンピュータシステムでリアルタイム処理機能を強化したオペレーティングシステム。処理要求の発生後，直ちにデータ処理を行い結果を返すことを主目的とする。

リアルタイム処理(real time processing)

コンピュータに処理要求の信号とデータを入力すると，ただちに処理し，出力や応答を行う処理方式。実時間処理，デマンド処理，即時処理ともいう。多くの場合，端末装置から通信回線を介して，センタのホストコンピュータにデータを入力し，処理結果の応答を受ける方式をとる。

リアルタイム制御システム(real time control system)

工業プロセスで，温度，圧力，流量などの物理量をリアルタイムで制御するシステム。プロセス制御(process control)システムともいう。センサなどを通じ装置を監視し，コンピュータでシステムを制御するには厳しい処理能力が必要で，コンピュータに処理要求の信号とデータを入力すると，ただちに処理し，出力や応答を行う。

リース(lease)

コンピュータやOA機器等の提供方法のひとつ。メーカや販売会社が製品をユーザに売るのではなく，一定の期間を設定して賃貸すること。コンピュータ関連製品は大規模になると高額であり，新製品も次々開発されることから，その使用にはリース方式が向いている。

リードオンリー属性(read only attribute)

MS-DOS，WindowsやUNIX系OSのファイル属性のひとつ。この属性をつけるとユーザは読出しや参照だけができ，ユーザコマンドではファイルの削除，更新ができない。保護したいファイルにつける。

リエンジニアリング(reengineering)

部品の再利用の際に，既存のシステムを調べ，新しい要求にかなうように変更すること。カスタマイズ(customize)が，既存システムの一部を修正することをさすのに対し，リエンジニアリングは，どこをどう直すべきかを知るための仕様分析の工程も含む。

リエントラント(reentrant)

→(参)再入可能プログラム

リカーシブ(再帰的)(recursive)

関数が自分自身をコールすること。リカーシブな呼出しは，自分自身を直接呼び出すことでも，他の関数経由で間接的にも実現できる。

リ

リスクマネージメント(risk management)

危険管理。情報処理の分野では，信頼性やセキュリティに関連したシステム全体の安全確保にともなう危険管理のこと。資産の価値とリスクの発生要因を分析して対策を講じ，最小のコストでリスクを最小に抑えることが目的である。

リスタート(restart)

コンピュータシステムで障害が発生した場合に，障害要因を取り除いた後で，障害発生の状態にシステムを復元してシステムを再度スタートすること。また，処理途中の状態になっているトランザクションがあれば，チェックポイントまで戻って(ロールバックして)再実行すること。リスタートが自動化されていれば，よりMTTRを低く押さえることができる。

リスト(list)

①プリンタなどに出力した，プログラム，データ，処理結果などの一覧表。プログラムリストの省略語。

②データ構造のリスト。→(参)リスト処理，リスト構造

リストア(restore)

磁気ディスクやフロッピーディスクに保存(バックアップ)しておいたデータをハードディスクに戻したり，システムに読み込むこと。操作上のミスやプログラムのバグによって，システムにトラブルが発生したときなど，元の状態に復旧することができる。→(関)バックアップ

リスト構造(list structure)

複数のデータがある順序にしたがって並んでいる構造。個々のデータの論理的な並びと主記憶装置内での並びが一致していない点で配列と異なる。順序を示す情報として各データはポインタを持つ。リスト構造のデータを操作・処理することをリスト処理という。代表的なリスト構造には次のようなものがある。

(1)単リスト(線形リスト)(linear list／linked list)

(2)双リスト(双方向リスト／連鎖リスト)(doubly-linked list)

(3)環状リスト(循環リスト／巡回リスト)(circular list)

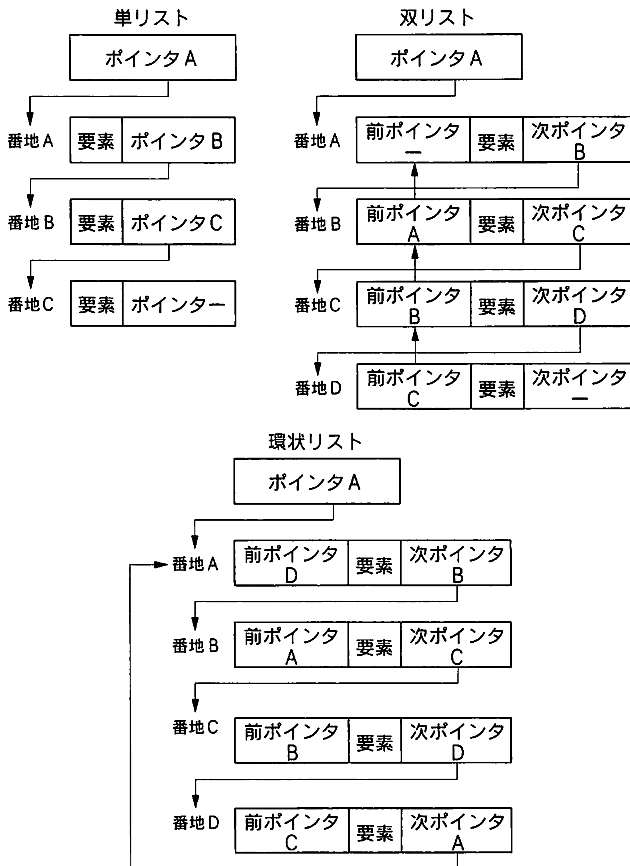
単リスト，双リストは要素の数が動的に変化する1次元の表である。単リストはアクセスをいっぽうの端に限るもので，スタックがこれに相当する。双リストは両端からのアクセスを許す。リスト構造を持ったデータ用のプログラム言語として，LISPやPROLOGなどがある。

リスト処理(list processing)



データ構造は配列とリストに大別できる。項目がある順序にしたがって並んでいる集合のことをリストという。リスト構造のデータを操作・処理することをリスト処理という。代表的なリスト構造に，単リスト(線形リスト)，双リスト(双方向リストまたは連鎖リスト)，環状リストなどがある。単リスト，双リストは要素の数が動的に変化する1次元の表である。単リストはアクセスをいっぽうの端に限るもので，スタックがこれに相当する。双リストは両端からのアクセスを許す。

リスト構造を持ったデータ用のプログラム言語として，LISPやPROLOGなどがある。



リセット(reset)

①ハードウェアを初期状態に戻すこと。ブートストラップをスタートさせコンピュータ内部の状態を元に戻し，改めて新しい仕事の準備をすること。

②レジスタや変数の値をデフォルト(初期値)に戻すこと。→(参)初期設定

リバースエンジニアリング(reverse engineering)

完成している製品を分析して，その源になる構造や設計方式を把握すること。さらに製作の工程を逆にさかのぼって得た情報を利用して，別の技術で類似製品やコンパチブル製品を作ること。解析複製ともいう。半導体やソフトウェア分野で広く行われている。

ソフトウェアの場合は，互換プログラムの開発が代表的ケース。プログラムの作成手順を逆にたどり，著作権法で保護されていない技術上のアイディアを抜き出して，自社のソフト開発に生かす手法である。著作権をめぐって係争問題になった例もある。

リファレンスデータベース(reference database)

情報源の案内を主たる目的とするデータベース(文献データベース)。データベースは当初，技術上の制約から情報がどこにあるのかを探す手段として始まり実用化に至った。リファレンスデータベースには文献のタイトルや著者名，掲載誌，内容の抄録などがある。今日では，検索した資料を直ちに入手できるファクトデータベースと連動するものが多い。

→(対)ファクトデータベース，(参)フルテキストデータベース

リプレース(replace)

使用していたコンピュータシステム等を新しいものに置き換えること。

リミットチェック(limit check)

入力データの内容をチェックする方法のひとつ。データの長さや値の上限や下限があらかじめわかっている場合に，その条件を満たしているかどうかを調べること。制限チェックチェックともいう。特にある範囲内にあるかどうかを調べることを範囲チェックという。

リモートジョブエントリ(remote job entry：RJE)

遠隔地の端末とセンタの端末を区別せず処理するセンタの計算機システムに，遠隔地の端末からジョブやデータを投入し，処理する方式。遠隔ジョブ入力ともいう。ジョブと入力データを一括してオンラインの端末装置から通信回線を介して入力し，ホストコンピュータがジョブ実行した結果をやはり通信回線を介して端末装置へ送り返す。データの送信や結果の入手に，郵送や輸送の必要がなくなるが，処理結果の受け取りにリアルタイム性は特に求めない方式といえる。

リモートバッチシステム(remote batch system)

遠隔地の端末とセンタのホストコンピュータのバッチ処理システムを通信回線で結んだ計算機システム。遠隔地の端末が送ってくるジョブやデータを，センタのジョブやデータと区別せずに，センタの計算機システムで処理する。プログラムやデータの送信や結果を入手に，郵送や輸送の必要がなくなり，入力データや出力結果は，通信回線を介して遠隔地の端末で入出力ができる。

リモートバッチ処理(remote batch processing)

オンライン処理の一形態。遠隔地の端末から通信回線を利用して，ホストコンピュータのバッチシステムに多量のデータを1度にコンピュータに伝送，入力して，処理を実行させ，結果を再びオンラインで返送する処理方式。遠隔一括処理ともいう。

リモートファイルアクセス(remote file access)

自システム内のファイルにアクセスするのと同様の操作で，ネットワークで接続した別のコンピュータのファイルを直接アクセス可能とする機能。実ファイルを持つ相手のコンピュータのことをファイルサーバと呼ぶことがある。UNIX，NetWare，Windows NTのネットワーク機能を持つOSで，この機能を提供している。遠隔ファイルアクセスの具体例には，NFS，RFS，VFSなどがある。

リモートログイン(remote login)

あるコンピュータから別のコンピュータ(ホスト)にログインすること。遠隔ログインともいう。この機能が利用できると，いながらにしてネットワークに接続されたどの利用可能なホストの端末(仮想端末)にもなれる。利用者のアカウントはネームサービス機能などによりあらかじめ利用するホストに登録しておく。利用者は，ホスト名を指定してログインする。ただし，これを無秩序に許すとクラッカー(ハッカー)の侵入を防御できないセキュリティホールになるおそれがある。

流通情報システム(physical distribution information system)

物流分野に適用した情報システムで，受発注業務，在庫照会，入出荷データ，貨物追跡管理，請求管理などのデータを電子化し，コンピュータや通信回線を利用し処理するシステム。物流EDIでは，荷主・倉庫・運送業者・金融業者など多数の事業者の相互間でのデータ交換が必要であり，標準化されたEDIプロトコルが使用されている。

利用者登録(registration of users)

情報資源のユーザであることを証明するために，権利者やオペレータのリストに権利の範囲や身分を記載すること。

(1)ハード・ソフトの購入者がメーカに対し行う登録は，バージョンアップや関連機器・ソフトの情報，メインテナンス情報等の提供を受けるためにも必要である。

(2)資源を共用するシステムでは，ユーザがシステムにアクセス権やID番号，パスワードなどを登録する。

利用者マニュアル(user's manual)

コンピュータシステムを使うユーザのための取扱い説明書。操作説明書ともいう。運用規定，手順，命令，操作法などのほか，エラー処理，トラブル処理の対処法なども含む。利用者には無縁の内部仕様などの説明は省き，利用の局面で必要な知識や技術を中心にまとめたもの。最近はマニュアルをハードディスク上に置き，オンラインマニュアルとして参照できるようにしてあるソフトも多い。→(類)マニュアル

リレーショナルデータベース(relational database：RDB)

関連するデータを，行と列に展開した複数の二次元の表形式で取り扱うデータベース。関係データベースともいう。人間が日常的に使う表の形式にイメージが近く，データの構造表現が柔軟で，データの独立性が高い点に特徴がある。表のデータを自由に取り出して加工でき，表の設計や拡張，変更が容易である。表の操作は集合論の関係代数を利用して演算で行う。

→(同)関係データベース，(関)関係代数

リンク(link)

①複数のオブジェクトモジュールを結合して実行可能なロードモジュールを作ること。実行前に必要なプログラムすべてを結合することを静的リンク，実行中に結合することを動的リンク(ダイナミックリンク)という。

②通信ネットワークのノード間をつなぐ通信路。

③データベースの索引用語の関連を示すときに使うコード。

リンクファイル(link file)

目的のファイルやフォルダへのパスを記録してリンクさせたアイコン。リンク先のファイルやフォルダを開くことができる。拡張子は.lnk。

リンケージエディタ(linkage editor)

→(同)連係編集プログラム

ル・レ・ロ

類義語検索(synonym retrieval／synonym search)

データベースや文書中の文字列データの検索を行うとき，類似性をもった用語まで範囲を広げて文字列を検索すること。例えば，アメリカを指定すると，USA，米国，合衆国などの用語も同時に検索すること。

ルーチン(routine)

一連の処理を行うプログラムのまとまり。

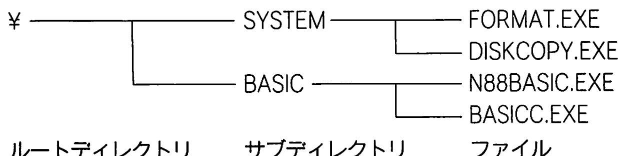
ルーチンワーク(routine work)

手順の定まった日常業務の意。→(同)定型業務，(対)非定型業務

ルートディレクトリ(root directory)

MS-DOSやUNIX系OSでは，ファイルを木構造のディレクトリによって管理している。データの管理を木(ツリー)に例えると，ファイルは葉(リーフ)に，ディレクトリは根(ルート)や節(ノード)に相当する。木構造の最上位(根に相当)のディレクトリをルートディレクトリという。ファイルが多い場合ルートディレクトリにすべてのファイルを保存すると煩雑になる。そこで，ルートディレクトリの下位に分岐ディレクトリ(サブディレクトリ)を作り，その中にファイルを分類，整理して保存する。さらに，そのディレクトリの下位に分岐ディレクトリを作ることもできる。

(例)MS-DOSの場合



ル

ループ(loop)

プログラム中で繰り返して実行する一連の命令。または，ある特定の条件を満たすまで繰り返し実行すること。繰り返し動作の打切り条件を正しく設定せず，プログラムが終了しない場合を永久ループ，無限ループ(permanent loop)，閉ループ(endlessloop)という。→(類)繰返し

ルックアンドフィール(look and feel)

OSやアプリケーションプログラムの操作ルールや画面デザイン。例えば，プロンプトマークの点滅で入力をうながすと決めたり，アイコンがプログラムやファイルの代替マークと決めたりすることなどの全体をさす。マウスのキーの操作法，ウインドウの操作法，アイコンの意味なども含む。アメリカのソフト会社間で画面や操作法の類似性をめぐって争われた裁判で，こうした操作ルールや画面デザインにもヒューマンインタフェースのくふうとして著作権が認められるようになった。

レイアウト(layout)

①ファイルに書込むレコードの属性やデータ長を考慮して，効率よく処理できるように設計すること。印字するデータについては，読みやすく理

解しやすい形式を設計すること。

②表示したり印刷する文書の文章や図表の配置。割付けともいう。近年はソフトの高機能化が進み，文字の書体，大きさや配色などを含んでレイアウトという。

レイアウト表示(display of layout)

ディスプレイ画面に当該ページのレイアウトを表示すること。ページ全体を縮小表示するが，文字や図版は簡略化して表示することが多い。

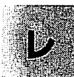
例外処理(exception handling)

通常の処理過程で発生する誤りや例外的な条件に対処するために用意した処理手順。対処すべき条件と，その条件が発生したときに実行する処理内容からなる。設定された条件が発生すると実行中の処理を行った後，例外発生の箇所に制御を戻し中断していた処理を継続する場合，制御を戻さずに安全な方向に処理の実行を終了する場合がある。

→(参)フェールセーフ，(参)フェールソフト

レグレッションテスト(regression test)

既存のシステムに修正，変更，機能追加をした場合，これらが正しく行われたことを確認するテスト。新機能部分だけでなく，従来機能のテストも行い，改変・追加したことにより，新たな誤り(レベルダウンという)が生じていないかを検証する。退行テスト，回帰テストともいう。



レコード(record)

相互に関連するデータ項目の集まり。データ処理は，レコードを単位として行う。論理レコードと物理レコードの二つに分ける。

①物理レコード(physical record)：入出力装置や補助記憶装置のファイルをアクセスするときに，実際の読み書きの物理的な単位として使用するデータの最小単位。一般に，関連するデータの集合が1論理レコードを形成し，複数の論理レコードをブロック化して 1物理レコードを構成し，複数の物理レコードの集合がファイルを形成する。

②論理レコード(logical record)：ファイルの構成要素で1件の事象に関連する項目(例えば個人情報なら，その個人の氏名・生年月日・性別・住所などのデータ)の集まり。プログラムでデータを取り扱う単位となる。

レコード型(record type)

E-Rモデルで，実体をある属性でグループ化した集合のこと。例えば，ある会社の社員であるという属性で作られた集合(社員ファイルに相当)は，実体(社員に相当)の属性データの並び(社員コード，氏名，生年月日，性別，所属など)といくつかの(社員数に相当)実現値の並び(9206123，合

格太郎，19720315，男，営業部など)で表現できる。COBOLのレコードに相当する。

レコードタイプ(record type)

ファイルに記録するレコードの形式。レコードの長さは，理論的にはいくらであってもよいが，実際にはどう扱うかによって処理の仕方が異なる。

(1)固定長レコード(fixed length record)

(2)可変長レコード(variable length record)

(3)不定長レコード(undefined length record)

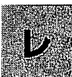
レコード長(record length)

レコードを形成する語または文字の長さ。

レジストリ(registry／registry data)

Windows95において，ハードウェアやソフトウェアに関する各種の設定情報を管理しているデータベース。Windows3.1ではいくつものINIファイルに設定情報が分散していたが，レジストリによって一元管理するようになった。レジストリはユーザごとの設定情報を管理するUSER.DATと，コンピュータごとの設定情報を管理するSYSTEM.DATの二つのファイルから構成されている。

②ハードやソフト，ネットワークなどの正当な使用者であることを登録すること，またはその登録簿に相当するデータベース。アクセス権の確認，課金情報の管理などに使用する。転じて，シェアウェアを使用する際に支払う寄付金など。



列(①column／field，②row，③sequence)

①表計算ソフトのワークシートで縦に並ぶ一連のセル領域。

②2次元配列や行列の縦方向の1列のこと。行(横に並ぶ1行)の対語。

③同一の型のデータが一列に並ぶデータ構造で，データの中身と順序が意味を持つもの。例えば，123と231は，10進数なら意味が異なる。列に対する操作には，参照，削除，挿入などがある。列を表現する代表的構造には，配列，リストなどがある。列のデータを操作できる場所を限れば，待ち行列やスタックになる。

レビュー(review)

もとの意味は，生徒が課題を再検討，復習すること，転じて指導者が批評，論評すること。システム開発では，システム設計者，プログラム作成者に，設計やコーディングの内容を記述，説明させ，あいまいな点や未熟

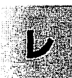
な点を探り，自覚させること。レビューの場は，一般には複数のメンバが参加し，作成者の報告や発表の後，問題点の指摘，批判を行い，再検討や改良を促し，後工程に誤りを伝えないようにする場でもある。このほか，メンバ間の情報伝達，教育，進捗管理のねらいもある。

レプリケーション(replication)

データベースを他のサーバ上で複製する機能。分散型データベースシステムでは，あるデータを特定のサーバにのみ置くと，ネットワーク上のある部分にトラフィックが集中したり，ネットワーク全体のトラフィック量が増加したりする。そこで，データベースの複製をいくつかのサイトで持ち，一番近くのサイトのデータベースにアクセスするようにすれば，ネットワーク上の負荷分散と負荷の低減が可能となる。なお，更新の際一時的に複製間の内容の不一致が発生するため，レプリケーションには不一致状態の管理と定期的に内容を一致させる機構が必要となる。

連係編集プログラム(linkage editor：LKED)

ユーザが書いたプログラムを実行可能なロードモジュールにするため，各種の計算機サブルーチンとユーザが作成したプログラムとを結合し編集するプログラム。結合編集プログラム，リンケージエディタともいう。ソースプログラムはコンパイラが機械語に翻訳しオブジェクトモジュールにする。一般に，オブジェクトモジュールは，アドレスの指定が0番地からとなっており，実行可能な形式ではない。また，他の副プログラムとのデータの授受の番地なども解決していない。実行可能にするためには，さらにさまざまな制御情報，呼び出される副プログラムや関数などの相互参照を解決し，可能ならば構成要素を再配置する必要がある。別々に翻訳したオブジェクトモジュールやロードモジュールから新たなロードモジュールを作成することが連係編集プログラムの役割である。



ローダ(loader)

①周辺装置や補助記憶装置からプログラムをコンピュータの主記憶領域に読み込ませるルーチン。常時メモリ内に存在している。

②指定したオブジェクトモジュールおよびロードモジュール(ユーザ用プログラム)を作り上げ，そのプログラムを実行させるサービスプログラム。特に，連係ローダ(linking loader)と呼ぶこともある。

ローディング(loading)

①入力装置にデータ媒体をかけること。例えば，磁気テープ記憶装置に磁気テープリールをかけ，テープを読取りヘッドにセットすること。

②プログラムを主記憶装置に入れること(プログラムローディング)。

ロード(load)

①主記憶領域のデータをアキュムレータやレジスタに転送すること。

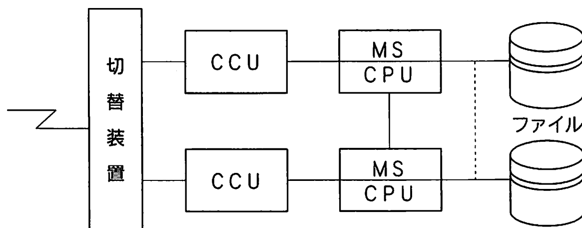
②プログラムやデータ等を補助記憶装置から取り出して，内部記憶装置に格納すること。

③入力装置にデータ媒体をセットすること。例えば，磁気テープ装置にリールを装着して，テープ始端マーカまで繰り出すこと。

④負荷，または処理要求。

ロードシェアシステム(load share system)

負荷分散(分割)システム。ある業務に関して複数台のコンピュータを使い，その処理形態(オンライン，バッチ)や処理発生地域などで区別し，負荷を分担させるシステム。銀行のオンラインサービスは地域による分担をしている。ひとつのシステムがダウンしても，他システムでカバーすることができる。



ロードファクタ(load factor)

直接編成ファイルでは，レコードのキー値とレコードアドレスは1対1で対応すれば理想であるが，一般的には，重複(シノニム)ができたり，逆に空き領域ができたりして記憶装置の使用効率が低下する。エクステントの大きさとレコード数の間には適当な比率がある。そこで予め最大レコード数を想定して，多対1の対応でアドレスを求めたうえ，さらに余裕を持たせるとシノニムの発生が抑制できる。このエクステントの最大記憶容量にたいする記録レコードの容量の比率をロードファクタという。約80%程度が適正である。→(類)記録密度

ロードモジュール(load module)

実行のために主記憶装置内へロードし得る計算機プログラムの単位。連係編集プログラムの出力がロードモジュールである。

ロールバック(roll back)

システム障害の発生により中断された処理を，その処理のすべてが行われなかった状態に戻すこと。マスタファイルのロールバックの場合，更新

前情報を用いてロールバック処理を行う。→$$

ロールバック(roll back)

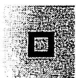
システム障害の発生で中断された処理を，その処理のすべてが行われなかった状態に戻すこと。マスタファイルのロールバックの場合，更新前情報を用いてロールバック処理を行い，この後ロールフォワード処理を行う。

ロールフォワード(roll forward)

ファイル装置の障害などで破壊されたファイルを，障害発生の直前の状態にまで復元させること。セーブしたファイル(バックアップファイル)を正常なファイル装置に戻し，セーブ時以降の更新後情報(ジャーナルログ)でファイルをすべて更新し，復元を行う。

ログ(log)

コンピュータシステムの動作，状態，操作状況などの時間的推移の記録。オンラインシステムなどで，どのような状態でシステムがダウンしても回復できるように，処理途中のファイルやデータベースの更新情報を記録しておくファイル。システムの稼働統計をとるときなどにも利用する。定期的にとる稼働記録をジャーナル(journal)といい，これらの記録をとることをロギング(logging)という。トランザクション処理では，ジャーナルの採取は必須である。



ロングファイルネーム(long file name)

255文字(半角)まで許容するファイル名。Windows95で使用できるファイル名の長さで，従来のMS-DOSのファイル名の長さが8文字と拡張子3文字に制限していたことに対比する語。

論理エラー(logic error)

プログラム命令の組立て方に誤りがあるために生じるエラー。シンタックスエラーに対比して用いる語。論理的な誤り(設計段階の誤解，プログラミング段階の誤記，実行段階の誤操作など)が原因となる。論理エラーの検出には各種のテストを行う必要がある。

論理データ設計(logical data design)

データフローダイアグラムや入出力仕様からサブシステム(モジュール)の内部に蓄積するデータの構造，項目，属性などを決定する工程。通常，次のような手順で行う。

(1)処理すべきデータ項目の検討

(2)データ間の関連性の分析，重複の排除，体系化

(3)ファイル仕様と媒体の決定

(4)上記の文書化

論理変数(logical variable)

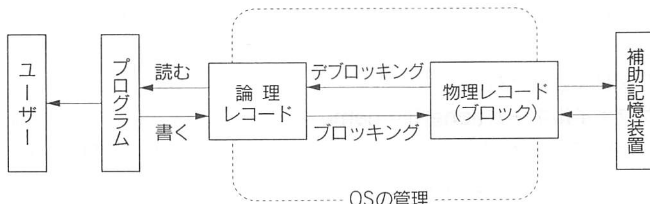
2値しか有しない論理演算のための変数。値は1か0，真か偽，HighかLowなど排他的である。ビットに特定の意味をつけてコード化した数(例えば0なら男性，1なら女性とか，0なら未婚，1なら既婚)は，データコードなので排他的であっても文字データと見なす。

論理レコード(logical operator)

プログラムから見た場合の入出力の単位。実際の入出力は，物理レコードを単位として行われる。通常，レコードといった場合は論理レコードをさす。→(対)物理レコード

論理レコード(logical record)

プログラムを作成する際に取り扱うひとつのまとまりを持った処理単位。例えば，個々の商品名と単位・売上数量からなる1件のレコードや，ある生徒に関する各教科の得点レコードなど。実際の入出力で論理レコードを何件かまとめて(ブロック化)，記憶することが多く，これは物理レコードという。



ワ

ワーキングディスク(working disk)

外部ソートなどの処理のために，一時的に作るファイルを記憶しておくための磁気ディスク。

ワーキングテープ(working tape)

分類，集計などの中間的な結果を仮に記憶するための磁気テープ。

ワークファイル(work file)

プログラムの翻訳，連係編集，データの分類等を行うとき，それぞれの中間的なプログラムや中間的なデータを一時的に外部記憶装置に保存して

おくための作業用ファイル。オペレーティングシステムが適当な名前をつけてファイル領域を確保し，使用後は自動的に削除する場合が多い。
→(類)スクラッチパドファイル，ワーキングディスク

ワークフロー(workflow)

業務の発生から終了までの仕事の流れ，または仕事の流れを決める規定や機構。ワークフローの改善で業務の効率化を行える。オフィスワークの中心が他部門との情報の交換，文書処理であることに注目し，情報の交換と処理にコンピュータネットワークを活用し，企業全体の生産性を高めようという構想に基づく。

ワークフロー管理(workflow management)

複数の人間が複数の工程で作業を行うとき，工程，負荷，設備などに偏りができないように計画すること。作業者に待ち時間ができたり，設備の空き時間が多いと全体としての効率は悪い。ワークフロー管理機能は，仕事の流れを監視し，管理上必要な情報を提供するものである。

ワード(word)

一般的には，ある目的をもつひとかたまりの文字列。

①ハードウェアの分野では，記憶装置，演算装置，制御装置などでひとまとめにして取り扱う情報の基本単位。計算機により長さ(ビット数)は異なり，制御用のマイクロプロセッサなどの4ビット単位のものから，64ビット単位までさまざまである。通常，1命令または1数値が1語，または2語に相当する。→(参)アドレス

②オペレーティングシステムが，処理の状態を管理するために常に保持している情報。



ワーニエ法(Warnier design method)

1970年代前半にJean-Dominique Warnierが提案したデータ構造設計法。データを集合としてとらえ，それが「どこで，いつ，何回」使われるかを明確化し，入力データの構造からプログラム構造を明らかにする。プログラム構造図は，入力データ構造の順次，選択，繰返しの表現としてそのまま反映される。LCP(logical construction program)ともいう。

ワーニングメッセージ(warning message)

コンピュータシステムからオペレータ，プログラマに出す応答メッセージ。プログラムの実行中，コンパイル中などに誤りの可能性がある状態を検出したとき，警告の意味で発するもの。→(参)エラーメッセージ

ワイヤフレームモデル(wire frame model)

3次元グラフィックで，物体をワイヤ(線素)の集まりで表す図形の表現

方法のひとつ。針金細工のように，物体を構成する面を囲む多数の線によって表現する。

ワイルドカード(wild card)

文字列やファイル名を検索する時に使用する，任意の文字や文字列の代わりを務める特別な文字。任意の1文字を表す「？」，任意の長さの文字列を表す「＊」などがある。

ワクチンプログラム(vaccine program)

コンピュータウイルスに侵されたプログラムやファイルを知らせたり，手当するプログラムの俗称。免疫プログラムともいう。ワクチンがウイルスに対抗する薬物であることから比喩的に名付けたもの。

割当て問題(assignment problem)

三つの事柄(例えば，材料の量と製品の量と販売利益／人と仕事内容と仕事量)の間に二つの重みの関係があるとき，問題で与えられた範囲内で，事柄を組み合わせ，課題の重みの総和を最小にする(反対の重みを考えれば最大にする)場合を求める問題のこと。例えば，与えられた材料で，何をどれだけ作るように割り当てるのが販売利益を最大にするか，与えられた人材を，何の仕事に何人割り振れば，最も多くの仕事ができるか(最も短期に一定量の仕事ができるか)という問題をさす。



数字

10進分類コード(decimal classification code)

10進数を用いるグループ分類コード。ある集合の要素を10個以下のグループに分類し，さらに各グループ内の集合の要素を10個以下のグループに分類し，各桁に数字を与えることを繰り返すコード化技法。代表的な例には図書分類のＮＤＣがある。

二項分布(binomial distribution)

離散分布のひとつ。例えば，不良率が$p(0 \leqq p \leqq 1)$である母集団からn個のサンプルを抜き取り，その中にx個の不良品が入っている確率が，

$${}_nC_x p^n (1-p)^{n-x}$$

となるような分布のこと。nを大きくすると正規分布で近似できる。

2次元の表(two-dimensional table)

要素が縦横に配置された行列状の表。2次元の表の基本項目を操作するには添字が2個必要である。

二次元配列(two-dimensional array)

配列のうち，二次元の配列のこと。

2層スキーマ(two schema structure)

CODASYL仕様のネットワーク型データベースでは，スキーマを2層でとらえ，論理ファイルの定義をサブスキーマ，物理ファイルの定義を記憶スキーマという。スキーマがデータベースの論理構造を全体として把握したもので，管理者の見方に近いのに対し，サブスキーマはスキーマの一部であって，適用業務プログラムにデータベースの構造を見せるプログラマの見方に近い。この役割をさらに分化させたのが3層スキーマである。

2分木(binary tree)

木構造の一つで，根(ルート)と呼ばれる根幹となる端点から二つ以下の枝がでるもの。各端点は節と呼ばれ，枝が0の端点は葉と呼ばれる。

2分探索(binary search／dichotomizing search)

2分木を利用した探索法。配列(リスト)中のレコードがキーの大小順(昇順または降順)に並んでいるときに効率のよい探索法の一つ。2等分探索法，2分割法ともいう。原理的には，レコードの並びのほぼ中央に位置するデータとキーの条件を比較して探索範囲を2分し，レコードの存在するはずのない一方を探索範囲から除外する。これを繰り返し少ない操作で効率よくデータを探す手法である。データの分布に偏りがあり，分布が既知の場合は，効率のよい別の分割法を用いることもある。AVL探索法やフィ

ボナッチ探索法などがある。

2分探索木(binary search tree)

2分木の頂点にレコードを入れ，右側の枝の先の頂点のすべてはこの頂点のデータより小さいものがあり，逆に左側には大きいデータがあるようにした木構造のこと。2分探索と同様の原理で，目的のデータを探すことができる。

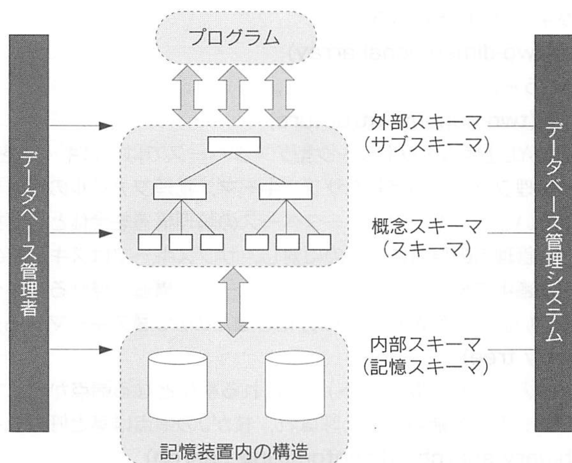
3層スキーマ(three schema structure)

ANSI／X3／SPARCが提唱した3層のスキーマ。

(1)概念スキーマ：コンピュータや利用分野などのことから離れて，事実の表現としてのデータの意味や関係の構造を記述するもの。

(2)外部スキーマ：データベース内のデータを操作する応用プログラムごとに，データの格納構造を記述するもの。

(3)内部スキーマ：概念スキーマと外部スキーマを，コンピュータや記憶装置上で実現するための物理的な構造を記述するもの。



8.3制限(8.3 file name)

MS-DOSのファイル名の付け方にある制限。ファイル名は英数字で8文字まで，拡張子は3文字という規格。Windowsになって，大幅に拡張された。→(参)ファイル名，ロングファイルネーム

英数字

ABC分析(ABC classification／Pareto's analysis)

在庫管理や品質管理に応用する重点管理項目を明らかにするための分析技法。ある特性に影響を与える要因が多くても，影響の程度は要因によって異なり，小数の要因が大きく影響している。このとき，最小のコストで最大の管理効果をあげるために，要因の重要度によってＡ，Ｂ，Ｃの3群に分け，Ａ群を重点的に管理する方法である。

例えば，1年間に使用する材料の金額をみると，金額的に多額のものは少しの点数であり，少額のものの点数は非常に多い。金額の大きい少品目を重点管理すれば少しの努力で大きな部分を管理することが可能となる。使用金額の大きい順に品目を並べる。累計金額を求めてグラフを作る。累計金額の合計を100％として，累計金額の約70～80％のものをA品目，70～95％をB品目，残りをC品目とし，A品目を重点管理する。ABC分析は在庫管理以外でも多くの場面に応用できる。

Ada(Ada)

1980年にアメリカの国防総省が開発した，組込み型計算機システムのための高水準プログラム言語。1988年，ISOで規格化し，1991年，JISも規格化した。PASCALの流れをくみ，型(type)の宣言に特徴がある。信頼性が高く，モジュール化したプログラム開発が容易で，リアルタイム制御システムなど大規模なソフトウェアの開発に適している。また，言語の設計と並行して，Adaによるプログラム開発の支援環境(Ada Programming Support Environment：APSE)も規定している。

API(Application Program Interface)

オペレーティングシステムとアプリケーションプログラムをつなぐインタフェース。アプリケーションプログラムがOSの機能を使うときに呼ぶ関数の集まり。OSがアプリケーションプログラムに提供するインタフェース仕様である。ハードウェアの機能を直接的に操作するアプリケーションプログラムを開発すると，同じOS用のアプリケーションプログラムでも機種が違うと動作しない場合がある。そこで採られたのが，OS側で設けた標準インタフェースであるAPIを介してハードウェアを操作するようにアプリケーションプログラムを開発する方式。APIを利用して開発したアプリケーションプログラムは，同じAPIを有する機種ならソースプログラムをコンパイルし直すだけで容易に移植できる。

ATM(automated teller machine)

銀行の窓口サービスを代行する自動支払機，自動預金機の両機能をもつ装置。自動窓口機ともいう。

AVI(Audio Visual Interleaved)

Windowsで動画および音声データを記録するためのファイル形式。Video for Windowsというソフトウェアを用いて再生する。なお，Windows95には同ソフトウェアが標準で組み込んである。

AVL木(AVL tree／Adel'son-Vel'skii-Landis tree)

平衡木の一種で，ある頂点から見て，左部分と右部分の枝の長さ(木の高さ，または深さともいう)の差が±1以内の木をさす。つまり，ある分岐点から見て左右の一方の枝のみが著しく長いということがない。2分探索木の探索の理想形は完全2分木であるが，平衡木であれば探索の効率は悪くない。そこで2分探索木のバランスが悪くなり，探索の効率も悪くなったとき，木の作り直し(組み替え)をする。このとき完全2分木でないAVL木のほうが計算量が少なくすむ。ただし，木の高さが大きくなることは避けられない。

B木(B-tree)

木構造の一種で，根からは1本以上，2m本以下の，各頂点からは最小m本(m≧3)，最大2m本の枝が出て，葉までの深さがすべて等しい木。これをm次のB木という。2分木に比較すると，ひとつの頂点から出る枝が多いので，データ量が多くなっても階層がさほど深くならない，平衡(バランス)をとりやすいという特色がある。

BASIC(Beginner's All purpose Symbolic Instruction Code)

大型コンピュータのTSS用の会話型言語として1964年に米国のダートマス大学のJ.G.KemenyとT.E.Kurtzらが開発した汎用プログラム言語。FORTRANに似た構造で，行列演算命令をもつのが特徴。

BMP(.BMP)

Windowsで標準的に用いる画像データのファイル形式。付属アクセサリのペイントツール等のお絵描きソフトを使って作成・加工・印刷することができ，デスクトップ画面の背景グラフィックス(壁紙)にも利用できる。

BOF(beginning of file)

ファイルの始まりを示すラベル。ファイルの情報を記録する。

BOT(beginning of tape)

磁気テープの始端を表すマーク。光を反射する金属の薄膜を使う。

BOV(beginning of volume)

ボリューム後書きラベル。ボリュームの先頭に記録した，収録するファ

イルの情報(内部ラベル)。ボリュームの内容を識別する。→(参)ラベル

C(programming language C)

システムやアプリケーション開発に用いる代表的なプログラム言語のひとつ。簡潔な表現で効率のよいプログラムが書け，関数定義による言語の拡張が可能という特徴をもち，異なるシステム間でも互換性が高い。

1973年，アメリカのベル研究所のD.RitchieらがPDP-11のOS，UNIXを開発するために設計したシステム記述用のプログラム言語。1975年に発表して以来，大型計算機からパーソナルコンピュータまで多くの処理系があらわれ，ハード依存性が弱いことから広く普及した。データ演算子の数が多く，簡潔な表現で効率のよいプログラムが書けること，ユーザ定義による言語の拡張が可能という特徴をもつ。

C++(C plus-plus)

アメリカのベル研究所(現在のAT&T)のB.Stroustrupが，1983年に開発したオブジェクト指向型のプログラム言語。C言語の仕様を含む拡張版。コンパイラやプリプロセッサのメーカにより仕様が多少異なる。

CAD(computer aided design)

コンピュータ支援設計。例えばLSIの設計，機械・建築・土木などの設計関係の作業をグラフィックディスプレイ装置，ディジタイザなどを用い，コンピュータと対話しながら行う。

CAE(computer aided engineering)

コンピュータ支援エンジニアリング。工業製品の設計段階で，技術計算，CAD，CAM，データベース，モデリングなどによりコンピュータを活用して計画，設計，解析，試作などを行うこと。
→(参)ファクトリオートメーション，(参)CIM



CAI(computer assisted instruction／computer aided instruction)

コンピュータ支援教育。コンピュータを利用した教授・学習システム。

CALS(commerce at light speed)

各種工業製品の開発，設計，受発注，流通等に必要な文書や図面などを，すべてコンピュータネットワーク上の電子データとして共有，交換するシステム。生産・調達・運用支援統合システムともいう。広い意味での電子商取引の基盤となることを目指している。1985年，米国国防総省の軍事物資受発注システムCALS(Computer-Aided Logistics Support systems)から始まった。その後，民生分野でも導入され，利用範囲と機能の拡張によりその概念も変遷してきた。1987年にはCALS(Computer-aided Acquisition and Logistics Support)，1993年にはCALS(Continuous Acquisition and Life-

cycle Support)に改称。

CAM(computer aided manufacture)

コンピュータ支援生産。例えば，機械加工等をコンピュータによって自動化して，無人化を可能にすることなど。代表的な例には，NC(numerical control)工作機械や産業用ロボットを制御して加工すること。数値制御系は受け取った数値を解釈して動作するシステムで，プラント制御や工作機械に応用する。

CASE(computer aided software engineering)

コンピュータ支援ソフトウェア工学。ソフトウェアの生産性を高めるため，要求分析，定義から設計，製造，保守の過程をコンピュータを利用して自動化する技術，工学。ソフトウェア工学が提唱する構造化の概念，開発方法論，各種の支援ツール(CASEツール)の統合などは，システムの大規模化，高度化，複雑化，開発期間の短縮，省力化の要求に応える。

CASEツール(CASE tool)

CASEの目的を実現するソフトウェア。システム開発の作業を支援するソフトには，従来からプログラミングツール，デバッグツール，テストツールなどがあった。最近は要求分析，定義，基本設計の段階(ウォータフォールモデルでいう上位工程)で用いる各種のツールも普及している。これらをCASEツールという。従来のツールと区別するときには，アッパーCASEツールともいう。また広義にはプログラミング支援ツールをも含めて，CASEツールと呼ぶことも多い。

CASL(CASL)

英数字

通商産業省が行っている情報処理技術者試験用のアセンブラ言語。COMETという仮想のコンピュータ上で動作する。CASLは4種類の疑似命令，3種類のマクロ命令，23種類の機械語命令からなる。1987年度から，それまで使ってきたCAP-Xに代わり，出題に使用している。

CD(cash dispenser)

銀行の窓口サービスを代行する現金自動支払機。

CIM(computer integrated manufacturing)

コンピュータ支援による統合生産管理。製造・生産における研究開発，設計，生産，販売，経営管理の各部門を統合して，省力化・自動化を目指す。基盤設備として，ネットワーク，データベース，分散コンピュータなどを使用する。統合CAD／CAMともいう。

CISC(complex instruction set computer)

多数の複雑な命令を処理できるCPUを搭載したコンピュータ。RISC(処

理できる命令数を減らすことで処理速度を速くしたCPUをもつ)に対比する語。RISCに比べ，単純な命令を実行する時も処理速度を短縮することができない。しかしアセンブラなどでプログラミングする場合，複雑な処理もひとつの命令で記述でき，人間側からは効率がよい。→(対)RISC

CODASYL型データベース(CODASYL type database)

CODASYLのDBTGが提案したモデルに基づいたネットワーク型データベース。構成要素や用語は処理系によって異なるが，基本的には，(1)スキーマ定義言語，(2)サブスキーマ定義言語，(3)データ操作言語，(4)データベース管理システムなどで構成される。1987年ISO規格となり，同年JIS規格のNDLとなった。

CPM(critical path method)

手順計画を矢線図で表し，各作業の直接費と所要時間の関係を直線で近似し，線形計画法を用いて費用最小の日程計画をもとめる手法。1957年にデュポン社が開発した。→(参)日程計画，クリティカルパス

CS(collaboration system)

もとは科学技術ネットワークの用語で，未解決の大きな問題の研究のために多数の研究者が協同して利用するデータベースやコンピュータ資源をさす。共有データベースには参加者が自分の実験結果や考察，意見などを書き込みメンバに公開し，討論の場ともなる。転じて，同様の協調的なグループの作業を支援するシステムをさす。

CSCW(computer supported cooperative work)

コンピュータ支援共同作業。LAN，ワークステーションなどの設備とグループウェアと呼ぶソフトウェアを利用して，主にオフィスにおける共同作業を行うこと。または，その環境。グループウェアの基本概念。具体的には，グループ参加者の個人スケジュールをコンピュータで管理した会議開催時間の最適スケジューリング，電子メールによる文書の回覧，閲覧，チェック，認証などを行う。グループウェアの適用形態として欧米で研究が進んでいる。→(関)グループウェア

CSV(comma separated value format)

アプリケーションソフトが扱うデータをテキストファイルとして入出力，記録する場合の形式で，フィールドとフィールドを半角のコンマ(，)で区切り，レコードとレコードを改行コードで区切る形式のこと。

CUI(character user interface)

ユーザがコンピュータを操作するとき，操作卓のキーボードからコマンドを入力して行う操作環境。多くのシステムで一般的な方式で，コマンド

に続いてオプションパラメタなどを指定するとシステムが内容を解釈して実行する。コマンドを熟知した人には便利な操作法。初心者にとってはコマンドによるシステム操作は負担が大きい場合もある。この問題を解消する目的でGUIの方式が考案された。→(対)GUI

CVFファイル(CVF file)

書き込み禁止，隠しファイル，システムファイルの属性をもつファイル。変更，削除するとシステムに異常をきたす重要なファイルをユーザの操作から遠ざけるためにCVFファイルとして特別に扱う。

D, E, F

DBMS(database management system)

→(同)データベース管理システム

DD／D(data dictionary／directory)

データベースではシステムが保持するデータの意味を定めるメタデータの記録のこと。データディレクトリは，データを扱うために必要な制御情報を集めたもので，システムやプログラムが使用する。いっぽう，データディクショナリは，人間が定義した管理情報である。DBMSは，これらの制御情報と管理情報をデータとして扱うさらに上位のメタデータをもつことができる。これらの情報によりシステムは，多様な機能をもち，データの複雑な操作ができるようになる。

DML(data manipulation language)

→(同)データ操作言語

EDI(electronic data interchange)

→(同)電子データ交換

EOB(end of block)

ブロック終結文字。ブロックの終わりを示す情報。→$$

EOF(end of file)

→(同)後書きラベル

EOS(electronic ordering system)

受発注業務を自動化したシステム。POSシステムと並び，商品の迅速な仕入れ，販売活動を支援する。顧客のニーズの多様化により，商品の多品種少量生産が進み，受発注業務が煩雑化する傾向にある。これをコンピュ

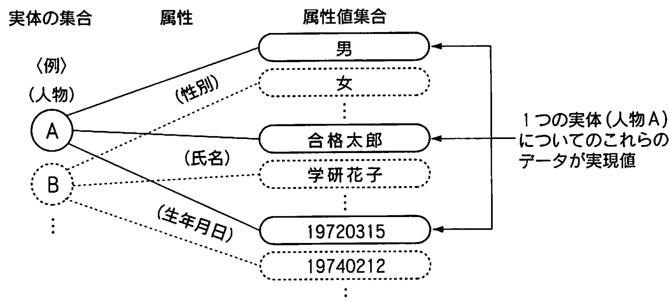
ータネットワークを利用し，簡便化，高速化を図るものである。

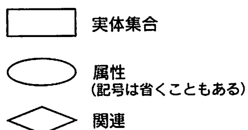
EOV(end of volume)

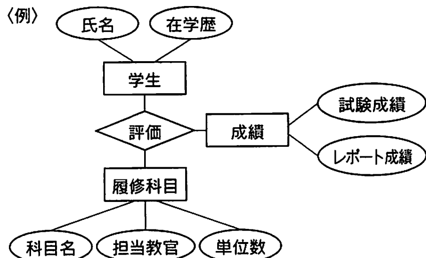
ボリューム後書きラベル。ボリューム中のデータの集合の後に置く内部ラベル。ボリュームの終わりを識別する。

E-Rモデル(entity relationship model)

データモデルの一種。現実の世界を実体(entity)と関連(relationship)という二つの概念を用いて表現するモデル。データベース設計のためのモデルとして適しており，広く普及した。例えば，ある人物(実体)は，(性別，氏名，生年月日)などの属性によって，許される属性値の集合の中から具体的な(男，合格太郎，19720315)などの実現値と関連づけられる。







複数の異なる実態集合の間には，関連があって結び付くことができ，例に示すような図記号で表すことができる。これをE-R図という。

FIFO(first in first out)

先入れ先出し。サービスや処理を受けるために，待ち行列(queue)があるとき，最初に入ってきたものから処理する方法。

FILO(first in last out)

先入れ後出し。後入れ先出し。サービスや処理を受けるために，スタッ

英数字

クがあるとき，最も後から入ったものからサービスや処理をする方法。LIFO(last in first out)ともいう。→(参)スタック，(同)LIFO，(対)FIFO

FLOPS(floating operations per second)

1秒間に処理できる浮動小数点演算の数。コンピュータの処理速度の目安として用いる。通常MFLOPS(メガフロップス)を単位とする。スーパコンピュータは，ギガFLOPS級の能力がある。

G, H, I

GIF(graphics interchange format)

米国のパソコン通信サービス会社コンピュサーブが考案したフルカラー静止画符号化方式。インターネットのWWWホームページのイメージデータ形式として標準的に使用したため有名になった。インターネット対応ソフトの多くにイメージのGIF変換機能やエディット機能が付加されている。256色までのカラーに対応できる。ファイルの拡張子は「.gif」。

GUI(graphical user interface)

ユーザがコンピュータを操作するとき，画面に表示しているメニューやアイコンをマウスなどで選択することで処理を容易に行えるようにした操作環境。従来の多くのシステムでは，コンピュータの操作はオペレータがコマンドをキーボードから入力する方式，すなわちキャラクタユーザインタフェース(character user interface：CUI)が一般的であった。GUIでは，画面上に機能や内容をイメージできる絵などを表示し，コマンドを知らなくても操作が類推できるなど優れた点が多い。ハードウェア，ソフトウェアに大きな負担がかかるため，処理スピード・容量などの点に問題があったが，近年の技術革新で高速なCPU，大容量のメモリが開発されたため，ユーザフレンドリなマンマシンインタフェースのひとつとして広く普及した。→(関)ルックアンドフィール

HDRラベル(header label)

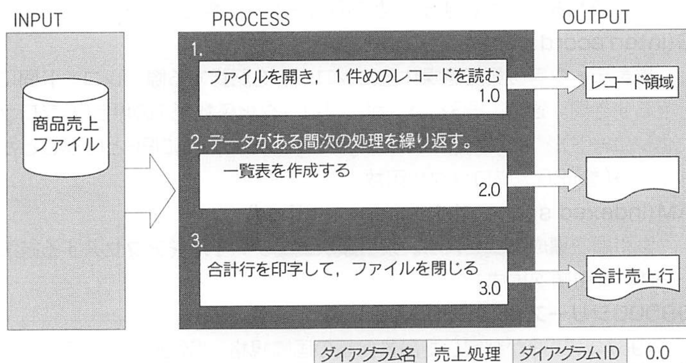
ファイル始めラベル／ファイル見出しラベル

HFS(hierarchy file system)

GUI環境のファイル管理システムにおける階層ディレクトリに相当する概念で，多重化したフォルダとその中身をマルチウィンドウで示すシステム。

HIPO(Hierarchy plus Input Process Output)

ソフトウェアの機能を，入力(input)・処理(process)・出力(output)に分けて階層構造に表示する図形化方式の手法。ソフトウェア開発時に，文書化の手法として，また，設計の補助手段として用いる。プログラム内のモジュールを木構造で表した内容一覧表，各モジュールの機能概要を書いた総括ダイアグラム，および機能詳細を書いた詳細ダイアグラムから成る。近年はCASEツールがとって代わりつつある。



HTML(hypertext markup language)

インターネットのWebページ記述用にスイスのCERNが開発したハイパーテキスト記述言語。言語仕様は，文書標準記述言語SGMLのサブセットになっている。HTMLは，参照文書，静止画，動画，音声などの入っているファイルを指定(タグ付け)できるようになっており，対応するページを画面に出力するとそれらがハイパーテキストとして表示できる。最近では，ワープロや表計算ソフトなどの出力結果をHTMLに自動変換し，そのままホームページにできるオフィスソフトが提供されている。

IBG(interblock gap)

→(同)ブロック間隔

ICカード(integrated circuit card)

ICメモリを内蔵したカード状の記憶媒体。磁気カード，フロッピーディスク(FD)に比べ記憶容量が大きく，書き換え可能などの利点も多く，磁気カードやFDに代わる新しい記憶媒体として実用化が進行中である。マイクロプロセッサを内蔵し，マイコン機能，インタフェース機能をもったカードもある。

ID(identifier／identification)

IDコードの略。利用者IDともいう。

IDカード(identification card)

①個人の資格や権利を識別する身分証明カード。広義にはクレジットカードなどを含む。→(関)IDカード

②コンピュータルームやシステムの自動旋錠機構(カードゲートシステム)に読み取らせる磁気カード，ICカード。入退室，ファイルアクセス等の際に資格を明らかにするためのIDコードを記憶している。

IRG(interrecord gap)

磁気テープ等磁気記憶媒体上にレコードを記録する際，レコード間にできるすき間。通常，IBGというが，ブロック化係数が1の場合(非ブロック化レコード)，1ブロック1レコードとなるため，特にIRGと呼ぶことがある。→(参)IBG，ブロック化因数

ISAM(indexed sequential access method)

索引順次編成アクセス法。索引順次編成ファイルをアクセスする技法。→(関)索引順次編成

ISO9000シリーズ(ISO9000 series)

1987年にISOが制定した品質保証の国際規格，ISO9001，9002，9003のこと。日本ではこれをもとに，1991年，JIS-Z-9900シリーズを規格化した。規格は，品質要求事項を満たすための品質システムの構築と品質マニュアルの作成を要求している。

J, K, L

JAN(Japanese Article Number)

日本バーコードシンボル規格

Java

アメリカのサンマイクロシステムズ社が1996年に開発したオブジェクト指向言語。1991年に携帯情報端末のアプリケーション記述用言語としてC＋＋をベースに開発したOak言語から発展した。Objective-CやSmalltalkなどの言語仕様も取込んでいるが，ほぼC＋＋同様にプログラミングができる。Javaはネットワーク上でダウンロードするアプリケーションソフトを開発するのに適した機能をもち，インターネット上で共通のプログラム

を実行するための標準的な言語になりつつある。ソースコードを中間コード(Javaアプレット)にコンパイルして実行するインタプリタ言語である。

JCL(job control language)

コンピュータシステムが能率よく稼働するために，制御プログラムに実行すべきプログラムを指定し，必要なコンピュータ資源(記憶域，入出力装置とデータ)の指定，競合がある場合の優先順位などを指示する言語。ジョブ制御言語ともいう。→(参)ジョブ

JPEG(joint photographic coding expert group)

①ISO／IEC JTCのSC29とITU-T共同でパッケージメディアの静止画像圧縮技術の標準化を進めている作業部会。

②同作業部会が定めたフルカラー静止画像のディジタル圧縮に関する標準。フルカラー静止画像符号化方式ともいう。

KE(knowledge engineering／knowledge engineer)

知識工学／知識工学技術者→(参)人工知能

KJ法(KJ method)

データ整理と分析の技法のひとつ。KJは，この手法を提案した川喜田二郎の頭文字。問題発見，原因分析，問題解決の過程で収集，観察，考察した情報をカードに記入し，これらをグループ化したり，相互関係を明らかにしたり，新たなカードをつくりながら，問題の全体像を考察する手法。記述型のモデル化手法であるが，多くの人の意見を取り入れられる点と適用範囲が広い点に特徴がある。具体的には，次のような手順と配慮の下に行う。

(1)1枚のカードには，複雑な要素を盛り込まず，簡潔な短い文を記入する。カードの枚数は多くてもよいが，表現は具体的であるほうがよい。

(2)記入のすんだ多数のカードを，まず直感的に共通性や関連性に注目してグループにまとめ，集めた理由を表札カードに記入する。

(3)小グループをさらにまとめ，表札カードを作る。これを繰返し，数グループになった時点で，カードを固定，枠線で囲む。相互関係を図解化する。

(4)グループ化の過程で考察した内容を文書に整理する。

KSDS(keyed sequential data set／keyed sequenced data set)

→(同)キー順データセット

LAN(local area network)

データ通信ネットワークの一形態で，同一敷地内，同一構内，同一建物

内など比較的狭い地域に分散して配置したコンピュータやワードプロセッサ，ファクシミリ，電話などで構成したネットワーク。

LHA

ファイルを圧縮し，使用時に解凍(伸張)するプログラム。フリーウェアとしてパソコン通信などで流通している。圧縮されたファイルにはLZHという拡張子がつく。吉崎栄泰の著作物。

LIFO(last in first out)

後入れ先出し法。広義には待ち行列の手法のひとつで，先に発生した事象から順に格納し，とり出すときは逆に最も新しく格納したものから順に取り出すこと。情報処理の分野では，例えばサブルーチンの戻り番地の管理や多重の割込みを処理する入れ子の構造のデータを処理するために，後入れ先出しの記憶装置(スタック)が必要になる。→(関)スタック

LISP(List Processor)

1960年代にマサチューセッツ工科大学のJ.McCarthyらが開発した代表的なリスト処理言語。リストとは，文章や数式や定理等の非数値データをうまく表す並びのことで，順序をもっている。リスト処理とは，これらの要素の並べかえ，追加，削除等の操作である。LISPは人工知能や計算論理学の世界で広く使う。→(関)リスト処理

LOGO(LOGO)

マサチューセッツ工科大学のS.Papertが開発した高水準の対話型プログラム言語。こどもがコンピュータで遊べるように設計してあり，パーソナルコンピュータによるCAI(コンピュータ支援学習)で多く用いる。ディスプレイ画面上のマークを動かして絵を描くタートルグラフィック機能(turtle graphics)をもつ。→(参)CAI

LP(linear programming)

オペレーションズリサーチ(OR)の技法のひとつ。線形計画法ともいう。

英数字

M, N, O

MICS

金融機関相互のネットワークのひとつで，CDやATMの相互利用を地域や業態を越えて可能にするシステム。全国キャッシュサービスともいう。加入機関の相互間で，現金引出し，残高照会が可能になる。1990年2月，

都市銀行のネットワーク(BANCS)と地方銀行のCDネットワーク(ACS)を結んでサービスを開始，翌年には，都銀，地銀，第二地銀協地銀，信金，信組，労金，農協の7つの業態の間で相互利用が可能になった。

Motif(Motif)

UNIXの標準化団体OSFの提唱するGUI環境。X-Windows上のユーザにGUI環境を提供する層で，ウィンドウズマネージャとツールキットから構成される。Motif上で開発されたアプリケーションは，画面構成や動作の仕方を一定の方式で作るため，利用者は新しいアプリケーションでも容易に利用できる。このように操作の仕方がすぐ理解できるデザインをルックアンドフィールといい，Motifでは，タイトルバー，メニューボタン，ウィンドウサイズの変更などが，共通の仕様になる。また画面全体をアイコン化できるため，画面を有効に使用することもできる。

MPEG(Motion Picture Expert Group)

ディジタル化すれば膨大になる動画像データを，CD-ROMなどの媒体に効率よく記録したり，ネットワークを使って伝送できるようにコンパクトに圧縮(エンコード)し，再生時に伸長(デコード)する技術の国際的な標準規格。現在はMPEG1とMPEG2の二つの規格を制定している。

(1)MPEG1はCD-ROMドライブでの再生を考慮しており，150キロbpsから最大1.5メガbpsのデータ転送速度で352×240ドットの画面サイズ，毎秒30コマの動画を再生することができる。画質的には家庭用VHS並みの品質である。

(2)MPEG2はハイビジョンクラスの高品位テレビにも対応し，アメリカやヨーロッパのディジタルTVでの採用が決まっている。データ転送速度や画素数によって何段階かのレベルに分かれている。

NC工作機械(numerical control machine tool)

NC制御による工作機械。加工情報や送り速度等を工作機械の情報処理部に紙テープ等で与え，これを操作し，工作物を加工する機械。NCボール盤，NC旋盤，NC中ぐり盤，NCフライス盤等がある。

NC制御(numerical control：NC)

数値情報によって制御すること。数値制御系は受け取った数値を解釈して，動作するシステムで，プラント制御や工作機械に応用する。

NDL(network database language)

ネットワーク構造のデータベースを定義・操作するデータベース言語のひとつ。NDLは，CODASYLのDBTG提案の言語仕様NDL(network database language)をもとにNDRが規定され，1987年にISO規格，JIS規格に

なった。次の3つのデータベース言語の構文と意味を規定している。

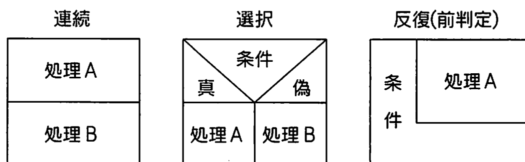
(1)スキーマ定義言語：DBの論理構造を定義する命令文の集まり。

(2)サブスキーマ定義言語：適用業務プログラムのビューによるDB部分の論理的定義

(3)モジュール言語とデータベース操作言語：適用業務プログラムがDBMSを利用するための定義

NSチャート(Nassi-Shneiderman chart)

T.NassiとB.Shneidermanが作った構造化プログラミング向きの処理手順の図式表現。連続・選択・反復構造を，図のように表す。



OEM(original equipment manufacturing)

相手先商標による製品の供給で，メーカが他社の開発，製造した製品を購入して，自社の商標で販売すること。

OLAP(online analytical processing)

エンドユーザが直接，基幹データベースを操作して問題点や解決策を見つけ出せるような分析型アプリケーションの概念。リレーショナルデータモデルを提唱したE.F.Coddが提唱した。クライアント／サーバ型，複雑な多次元データ構造を処理可能，データベースの物理構造を意識する必要がない，同時アクセスができかつネットワークで分散したデータベースをリンクして高速に検索できるなどの特徴をもつ。経営者の意思決定支援システムや役員情報システムなどの構築に向く。なお，OLAPを構築できるシステム環境をデータウェアハウスと呼ぶ。

OLTP(online transaction processing)

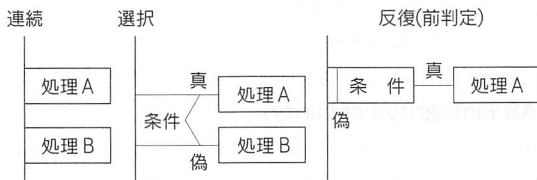
オンライントランザクション処理。

OR(operations research)

→(同)オペレーションズリサーチ

PAD(problem analysis diagram)

1970年代後半に考案した処理手順の図式化手法のひとつ。基本制御構造は次のように表現する。問題分析図ともいう。

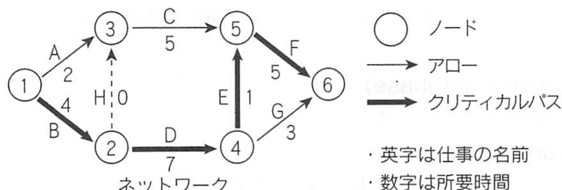


PDS(public domain software)

→(同)パブリックドメインソフト

PERT(program evaluation and review technique)

日程管理や工程管理に使う図式的な手法。システム開発や工事の実施計画などを組む場合，どのような方法で，どのような工程の進め方をしたら人員・資材・時間のむだがないように配置でき，工期を短縮できるかを計画立案する。目標を達成するために必要な諸活動を時間的に定義し，作業の進行状況を管理する手法。



POSシステム(point of sales system)

電子式金銭レジスタ，バーコード読取り装置，クレジットカードの自動判別装置などの機器により収集した，商品の販売情報などをコンピュータに送り，情報の有効利用を図るためのシステム。販売時点情報管理システムともいう。初期のシステムは，レジスタで集計したものをフロッピーディスクに格納してセンタへ配送していた。最近では，回線で直結し，商品が購入されるごとにそのデータをセンタへ即時に送る。そのため，現在店頭で何が売り切れそうかを監視することが可能で配送に無駄がなくなる。

QC(quality control)

品質管理。買い手の要求にあった品質の製品を経済的に作り出すすべて

の手段の体系。主な手法に，管理図法，抜取検査法，実験計画法がある。

QC七つ道具(seven tools for quality control)

近代的な品質管理の手法「事実に基づいた管理」を具現化する基本的・代表的な手法のこと。当初は製造部門(工場)の品質管理から生まれた。「七つ道具」は「ひとそろい」程度の意味で，時代や適用部門で異なってもよい。近年は「新QC七つ道具」の提案もある。

(1)パレート図，(2)特性要因図，(3)ヒストグラム，(4)グラフ，(5)チェックシート，(6)散布図，(7)管理図，(8)層別

RASIS(RAS+integrity+security)

コンピュータシステムのハードウェア・ソフトウェア両面における信頼性の概念RASに，Integrity(完全性)とSecurity(機密性・安全性)を加え，より高い信頼性の技術を総称したもの。

RCS(remote computing service)

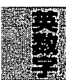
コンピュータの計算サービスを行っている計算センタなどが，一括処理ではなく，端末を利用者のところに置きオンラインによってそのサービスを提供するもの。利用者は，データを計算センタへ持ち運ぶ必要がない。

RDA(remote database access)

遠隔地にあるデータベースをアクセスするためのアプリケーションインタフェースのOSI標準規格。標準アクセス言語はSQL。データベースサーバ側とクライアント側のアクセス手順が障害発生時のやり取りも含めて規定している。

RDB(relational database)

→(同)関係データベース



RDF(record define field)

レコード定義フィールド。→(参)制御インタバル

RISC(reduced instruction set computer)

CISCコンピュータに比して，厳選した少ない命令体系をもつコンピュータ。RISCはVLSIに適したアーキテクチャで，ワークステーション，パソコンのCPUに採用され，現在の主流になりつつある。縮小命令セットコンピュータともいう。CISCコンピュータのプログラムで命令の使用頻度を調べたところ，大半が転送命令等の簡単でサイクルタイムの短いものであった。そこであまり使用しない複雑な命令を減らしアドレス方式を単純にすることでハードを簡素化し，高速化するコンピュータRISCが登場した。複雑な命令はコンパイラで簡単な命令の組合せにコード化する。したがって，一般にオブジェクトプログラムはCISCより大きくなる。

RRDS(relative record data set)

→(同)相対レコードデータセット

RRN(relative record number)

→(同)相対レコード番号

SE(systems engineer)

→(同)システムエンジニア

S, T, U

SGML(Standard Generalized Markup Language)

電子文書交換のためにISOが標準化した汎用の文書定義言語。テキストに文字種，表示，印字の体裁まで含めて記述できるほか，見出し，本文，索引など文書の構造を定義できる。ISO／IEC JTC1のSC18が定め，1986年にISO規格となった文書技術言語。標準一般化マーク付け言語ともいう。SGML規約に準拠して文書を記述するとマルチメディアを意識したハイパーテキストの文書の検索，改訂，製本，版管理が容易になる。複雑な機械類の開発を委託するような場合に，SGML文書にすることにより，文書の取り扱いの容易さと正確性が増大する。そのため，CALSのようなコンピュータの支援を受けた資材取引にSGML文書の利用を検討している。HTMLは，SGMLを拡張し，ページ間のリンクの指定やデータ入力フィールドを指定できる。→(参)HTML

SI(system integration)

→(同)システムインテグレーション

SOHO(small office-home office)

小規模事業所，個人住宅内で行うビジネス，または労働形態を象徴する語。1990年代後半から大規模・集中型の企業形態，労働形態に対比する合理化，効率化の概念として登場した。

SQL(structured query language)

データベースに対する操作言語の国際標準。1987年にISOが国際標準とした。同年JIS規格にもなる。最近の情報処理試験でも基本的なSQL文を理解しているかどうかを問う問題も出題されている。単なるデータの照会や更新だけでなく，データベース自体の構造の拡張，変更も仕様に含む。市販のDBMS製品はほとんどSQLに準拠している。また，分散型データベー

スの検索も想定している点がSQLの大きな特徴でもある。

STS分割(Source／Transform／Sink division)

→(同)源泉／変換／吸収分割

Tiff(Tagged Image File Format)

画像データを記録するファイル形式。異なる機種やアプリケーション間で画像データの互換性を実現することを目的として，マイクロソフト社とアルダス社が中心となって規格をとりまとめた。単純なモノクロ画像からグレースケール，1677万色のフルカラー画像に至るまで，さまざまなグラフィックス形式に対応しており，写真などのスキャニングした画像を扱うのにも適している。タグと呼ばれる説明部分に，データをどんな形式で記録したのかを書き込む。

TSS(time sharing system)

→(同)タイムシェアリングシステム

UNIX(UNIX)

小型コンピュータ用のオペレーティングシステムのひとつ。ミニコンピュータからパーソナルコンピュータまでの共通利用や異機種間のネットワークを利用する上ですぐれた機能をもち，現在はワークステーション級のコンピュータの主流オペレーティングシステムになっている。タイムシェアリングシステム用のオペレーティングシステムで，マルチユーザマルチタスク(同時に複数の利用者が複数の仕事を行う)向けになっている。

V, W, X, Y, Z

VAN(value added network)

第一種電気通信事業者が提供する電話網，テレックス網などを借り受け，その基本伝送サービスに加えて，異機種コンピュータ相互の接続サービス(プロトコル交換，フォーマット変換，コード変換，メディア変換)，メールボックス機能，データベースなどの通信処理サービス，情報処理サービスを提供するネットワーク。付加価値通信網ともいう。VANには，(1)特別第二種電気通信事業者が提供する汎用VANと(2)一般第二種電気通信事業者が提供する特別VANがある。

VSAM(virtual storage access method)

仮想記憶のもとで，索引順次編成(ISAM)に相当するデータセットを直接

アクセスする方法。旧来のデータ管理が提供している順アクセスの機能，直接アクセスの機能をひとつにまとめたアクセス法で，装置からの独立性が高い。VSAMでアクセスするデータセットはVSAM特有な構造をしている。VSAMデータセットに対して，順，直接あるいはスキップ順にデータの読み込み，書き出し，追加，交信，削除を行うことができる。

VSAMデータセットにはエントリ順データセット(Entry Sequenced data Set：ESDS)，キー順データセット(Key Sequenced data Set：KSDS)，相対レコードデータセット(Relative Record data Set：RRDS)がある。

VSAMファイル(virtual storage access file)

仮想記憶のもとで，索引順次編成(ISAM)に相当するデータセットを直接アクセスすることのできるファイル。記憶形式が直接アクセス装置のタイプに依存しない，記憶形式，アクセスの特徴に応じたデータセットが選択できるなどがVSAMの特徴である。

WAVEファイル(WAVE file)

音声データを記録するためのファイル形式。PCM方式によってデジタル化した音声を録音，再生する。Windowsでは標準の形式である。Macintoshでの再生には，SoundEditのような再生ソフトを使用する。

WYSIWYG(what you see is what you get)

ワードプロセッサや帳票作成プログラムの処理中，ディスプレイ画面上に表示した文書や帳票を，そのままの形式でプリントアウトできる機能。「目で見たものが手に入る」というスローガンの頭文字を並べたもの。ユーザにとっては便利な機能である。

zip(zip disk)

zipは，富士写真フィルムが開発した書き換え可能な磁気ディスクで，プラスティック製カートリッジ入りのリムーバブルメディア。直径90mmで約100メガバイトの記憶容量がある。zipドライブは，米国アイオメガ社，日本ではセイコーエプソンが開発した。平均データ転送速度1.25メガバイト／秒で，ほぼHDDと同速度でアクセスする。



通産省情報処理技術者試験　シスアド・第二種・第一種受験対策

合格情報処理8月号第2付録

情報システム用語辞典(五十音順・1998年度版)

1998年8月1日発行

発行人　阿部庄之助

編集人　忍足惠一

発行所　株式会社　学習研究社

印刷所　共同印刷株式会社

☆この雑誌に関するお問い合わせなどがありましたら，次のところにご連絡ください。

〒145-8502　東京都大田区仲池上1-17-15　学研　お客さま相談センター「合格情報処理」係

・電話は，内容について → 03-5742-2601，FAX　03-5742-2618

その他は → 03-3726-8124(お客様相談センター)